



**TUGAS AKHIR - TE141599**

**PERANCANGAN SISTEM PENGATURAN ADAPTIF UNTUK  
*TRAJECTORY TRACKING QUADCOPTER* PADA GERAK  
*CRUISE***

Benaverd Rizanda Putra Leihitu  
NRP 2212100157

Dosen Pembimbing  
Ir. Rusdhianto Effendie A.K., MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



***FINAL PROJECT - TE141599***

***QUADCOPTER'S TRAJECTORY TRACKING ADAPTIVE  
CONTROL DESIGN FOR CRUISING MOVEMENT***

Benaverd Rizanda Putra Leihitu  
NRP 2212100157

*Supervisor*  
Ir. Rusdhianto Effendie A.K., MT.

***ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017***

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Perancangan Sistem Pengaturan Adaptif Untuk *Trajectory Tracking Quadcopter* Pada Gerak *Cruise*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 5 Januari 2017

Benaverd Rizanda Putra Leihitu  
NRP 2212100157

*--halaman ini sengaja dikosongkan--*

**PERANCANGAN SISTEM PENGATURAN ADAPTIF UNTUK  
TRACKING QUADCOPTER PADA GERAK  
CRUISE**

**TUGAS AKHIR**

**Dibuatkan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Teknik Sistem Pengaturan  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Menyetujui:**

**Dosen Pembimbing**



**Ir. Rusdhianto Effendie A.K., MT.**

**NIP. 195704241985021001**



-- *Halaman Ini Sengaja Dikosongkan* --

# **PERANCANGAN SISTEM PENGATURAN ADAPTIF UNTUK TRAJECTORY TRACKING *QUADCOPTER* PADA GERAK *CRUISE***

**Nama** : Benaverd Rizanda Putra Leihitu  
**NRP** : 2212100157  
**Pembimbing** : Ir. Rusdhianto Effendie A.K., MT.  
**NIP** : 195704241985021001

## **ABSTRAK**

Gerakan *cruise* pada *quadcopter* dapat dilakukan bila *quadcopter* memiliki kestabilan dalam melakukan gerak rotasi dan gerak translasi. Dalam simulasi diusahakan respon gerak rotasi lebih cepat daripada respon gerak translasi. Pada *quadcopter* ini, perubahan sudut *pitch* mempengaruhi posisi *quadcopter* pada sumbu X, sedangkan perubahan sudut *roll* mempengaruhi posisi *quadcopter* pada posisi Y. Selain itu *quadcopter* memiliki system yang *non-linear* dan memiliki kestabilan yang rentan terhadap gangguan. Pada penelitian Tugas Akhir ini dirancang pengendalian gerak *cruise quadcopter* menggunakan kontroler *Proportioinal Derivative* (PD) untuk gerak rotasi dan kontroler PD Adaptif untuk pengendalian gerak translasi. Untuk mendapatkan parameter dari PD dan PD Adaptif digunakan metode *MIT Procedures*.

**Kata Kunci:** *Quadcopter, Cruise, Proportional Derivative, Adaptive Control*

*--halaman ini sengaja dikosongkan--*



## ***QUADCOPTER'S TRAJECTORY TRACKING ADAPTIVE CONTROL DESIGN FOR CRUISING MOVEMENT***

***Name*** : Benaverd Rizanda Putra Leihitu  
***Student Number*** : 2212100157  
***Supervisor*** : Ir. Rusdhianto Effendie A.K., MT.  
***ID Number*** : 195704241985021001

### ***ABSTRACT***

*Quadcopter's cruising movement requires the stability of its rotational and translational movement. It also requires the respond of rotational movement to be rapider than the respond of translational movement. Quadcopter's changes at pitch angle is affecting its position at axis-X, and quadcopter's changes at row angle is affecting its position at axis-Y. Quadcopter is a non-linear plant, so it is very susceptible to be affected by noise or disturbances. The purpose of this final project is to make a new controller design for quadcopter's trajectory tracking for cruising movement. This final projet uses proportional derivative (PD) controller for the rotational movement and adaptive PD controller for the translational movement. The parameters of both PD and adaptive PD are determined through MIT Procedures Method.*

***Keywords:*** *Quadcopter, Cruise, Proportional Derivative, Adaptive Control*

*--halaman ini sengaja dikosongkan--*

## KATA PENGANTAR

*Assalamualaikum wr. wb.*

Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan buku Tugas Akhir ini. Shawat serta salam senantiasa tercurah pula kepada nabi besar baginda Rasulullah Muhammad SAW.

Buku Tugas Akhir ini disusun untuk melengkapi salah satu syarat memperoleh gelar sarjana teknik di jurusan Teknik Elektro ITS. Buku yang berjudul ***“Perancangan Sistem Pengaturan Adaptif Untuk Trajectory Tracking Quadcopter Pada Gerak Cruise”*** dipersembahkan juga untuk kemajuan riset dan teknologi Indonesia khususnya untuk ITS, Fakultas Teknologi Industri, Jurusan Teknik Elektro, dan bidang studi Teknik Sistem Pengaturan.

Pada kesempatan ini tak lupa penulis menyampaikan rasa terima kasih kepada beberapa pihak, antara lain:

1. Allah SWT yang telah memlimpahkan ramhat dan hidayah-Nya serta memperlancar dalam pengerjaan Tugas Akhir ini.
2. Orang tua dan kedua adik saya yang secara tidak langsung menjadi sumber semangat dalam menyelesaikan Tugas Akhir ini.
3. Dosen Pembimbing, Bapak Ir. Rusdhianto Effendie A.K., MT. atas segala bimbingannya kepada penulis dalam pengerjaan Tugas Akhir ini.
4. Semua dosen Jurusan Elektro atas segala ilmu dan masukan yang diberikan selama menimba ilmu disini.
5. Bapak dan Ibu penguji Tugas Akhir yang telah memberi masukan kepada penulis sehingga buku ini menjadi lebih baik.
6. Sahabat-sahabat di kosan tercinta, Wisnu, Adit, Bandung, Igun, Oya, dan teman-teman lain yang tidak saya sebutkan.
7. Teman-teman SAT Project, Ozie, Adit, Hafidh, Icing, Raihan, Ican, yang telah menemani saya memenuhi hobi bermain music di Surabaya.
8. Teman-teman angkatan 2012 Teknik Elektro ITS, khususnya bidang studi Teknik Sistem Pengatuan yang telah berbagi suka dan duka.
9. Semua pihak yang turut membantu dalam pengerjaan Tugas Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari dan memohon maaf karena masih banyak kekurangan pada Tugas Akhir ini. Kritik dan saran selalu penulis nantikan agar menjadi lebih baik pada masa mendatang. Akhir kata, penulis berharap Tugas Akhir ini dapat bermanfaat dan menjadi acuan dalam penelitian selanjutnya.

Surabaya, Januari 2017

Penulis

# DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>PERNYATAAN KEASLIAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vii</b>
<b>ABSTACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>DAFTAR ISI .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xv</b>
<b>DAFTAR TABEL .....</b>	<b>xvii</b>
 <b>BAB I PENDAHULUAN .....</b>	 <b>1</b>
1.1 Latar Belakang.....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan .....	4
1.7 Relevansi .....	5
 <b>BAB II TEORI PENUNJANG .....</b>	 <b>7</b>
2.1 <i>Quadcopter</i> .....	7
2.1.1 Konsep Dasar <i>Quadcopter</i> .....	7
2.1.2 Kinematika <i>Quadcopter</i> .....	11
2.1.3 Dinamika <i>Quadcopter</i> .....	17
2.1.4 Model Matematika.....	24
2.2 Kontroler.....	25
2.2.1 <i>Proportional Derivative</i> (PD).....	27
2.2.2 Kontroler Adaptif.....	28
2.3 Linierisasi .....	30
 <b>BAB III PERANCANGAN SISTEM .....</b>	 <b>33</b>
3.1 Parameter <i>Quadcopter</i> .....	33
3.2 Linierisasi Persamaan <i>State Quadcopter</i> .....	33
3.2.1 Linierisasi Persamaan Gerak Rotasi .....	34
3.2.2 Linierisasi Persamaan Gerak Translasi .....	35
3.3 Perancangan Kontroler PD .....	37
3.3.1 Perancangan Kontroler PD Sudut <i>Roll</i> .....	37

3.3.2	Perancangan Kontroler PD Sudut <i>Pitch</i> .....	38
3.3.3	Perancangan Kontroler PD Sudut <i>Pitch</i> .....	39
3.3.4	Perancangan Kontroler PD Gerak <i>Heve</i> sumbu-Z ..	40
3.4	Perancangan Kontroler PD Adaptif .....	40
<b>BAB IV</b>	<b>PENGUJIAN DAN ANALISIS .....</b>	<b>45</b>
4.1	Skema Simulasi Pergerakan <i>Tracking Quadcopter</i> .....	45
4.2	Simulasi Tanpa Noise .....	46
4.2.1	Simulasi Stabilisasi Gerak Rotasi <i>Quadcopter</i> .....	47
4.2.2	<i>Trajectory Tracking Quadcopter</i> Tanpa Noise .....	48
4.3	Simulasi Pengujian dengan <i>Noise</i> .....	53
4.3.1	<i>Trajectory Tracking Quadcopter</i> Dengan <i>Noise</i> Konstan .....	54
4.3.2	<i>Trajectory Tracking Quadcopter</i> Dengan <i>Random</i> <i>Noise</i> .....	59
<b>BAB V</b>	<b>PENUTUP .....</b>	<b>65</b>
5.1	Kesimpulan .....	65
5.2	Saran .....	66
<b>DAFTAR PUSTAKA.....</b>		<b>67</b>
<b>LAMPIRAN A .....</b>		<b>A1</b>
A.1	Simulink Keseluruhan Sistem.....	A1
A.2	Simulink Pengendalian Sudut <i>Roll</i> .....	A1
A.3	Simulink Pengendalian Sudut <i>Pitch</i> .....	A1
A.4	Simulink Pengendalian Sudut <i>Yaw</i> .....	A2
A.5	Simulink Pengendalian Posisi X .....	A2
A.6	Simulink Pengendalian Posisi Y .....	A2
A.7	Simulink Pengendalian Posisi Z .....	A2
A.8	Simulink <i>Switch Manual</i> untuk <i>Noise</i> .....	A3
A.10	Simulink Linierisasi Gerak Rotasi .....	A4
A.11	Simulink Linierisasi Gerak Translasi.....	A5
<b>LAMPIRAN B.....</b>		<b>B1</b>
B.1	Program PD Adaptif Posisi X Matlab.....	B1
B.2	Program PD Adaptif Posisi Y Matlab.....	B1
<b>RIWAYAT PENULIS .....</b>		<b>C1</b>

## **TABLE OF CONTENTS**

<b>TITLE .....</b>	<b>i</b>
<b>AUTHENTICITY .....</b>	<b>iii</b>
<b>APPROVAL SHEET .....</b>	<b>v</b>
<b>ABSTRACT (INDONESIA) .....</b>	<b>vii</b>
<b>ABSTRACT (ENGLISH) .....</b>	<b>ix</b>
<b>PREFACE .....</b>	<b>xi</b>
<b>TABLE OF CONTENTS .....</b>	<b>xiii</b>
<b>LIST OF FIGURES .....</b>	<b>xv</b>
<b>LIST OF TABLES .....</b>	<b>xvii</b>
 <b>CHAPTER I INTRODUCTION .....</b>	 <b>1</b>
1.1 <i>Background</i> .....	1
1.2 <i>Problem Formulation</i> .....	2
1.3 <i>Boundary Problem</i> .....	2
1.4 <i>Objective Research</i> .....	3
1.5 <i>Methodology</i> .....	3
1.6 <i>Writing Systematic</i> .....	4
1.7 <i>Relevance</i> .....	5
 <b>CHAPTER II BASIC THEORY .....</b>	 <b>7</b>
2.1 <i>Quadcopter</i> .....	7
2.1.1 <i>Quadcopter's Basic Theory</i> .....	7
2.1.2 <i>Quadcopter's Kinematics</i> .....	11
2.1.3 <i>Quadcopter's Dynamics</i> .....	17
2.1.4 <i>Mathematic Model</i> .....	24
2.2 <i>Controller</i> .....	25
2.2.1 <i>Proportional Derivative (PD)</i> .....	27
2.2.2 <i>Adaptive Controller</i> .....	28
2.3 <i>Linearization</i> .....	30
 <b>CHAPTER III SYSTEM DESIGN .....</b>	 <b>33</b>
3.1 <i>Quadcopter's Parameter</i> .....	33
3.2 <i>Quadcopter's State Equation Linearization</i> .....	33
3.2.1 <i>Rotational Movement Equation Linearization</i> .....	34
3.2.2 <i>Translational Movement Equation Linearization</i> ...	35
3.3 <i>PD Controller Design</i> .....	37
3.3.1 <i>Roll Angle's PD Controller Design</i> .....	37

3.3.2	<i>Pitch Angle's PD Controller Design</i> .....	38
3.3.3	<i>Yaw Angle's PD Controller Design</i> .....	39
3.3.4	<i>Heve Movement PD Controller Design</i> .....	40
3.4	<i>Adaptive PD Controller Design</i> .....	40
<b>CHAPTER IV SIMULATION RESULT .....</b>		<b>45</b>
4.1	<i>Quadcopter Tracking Movement Simulation's Scheme</i> ...	45
4.2	<i>Simulation Without Noise</i> .....	46
4.2.1	<i>Quadcopter's Rotational Stabilization Simulation</i> .	47
4.2.2	<i>Trajectory Tracking Quadcopter Without Noise</i> .....	48
4.3	<i>Simulation With Noise</i> .....	53
4.3.1	<i>Trajectory Tracking Quadcopter With Constant Noise</i> .....	54
4.3.2	<i>Trajectory Tracking Quadcopter With Random Noise</i> .....	59
<b>CHAPTER V CLOSING .....</b>		<b>65</b>
5.1	<i>Conclusions</i> .....	65
5.2	<i>Advices</i> .....	66
<b>BIBLIOGRAPHY .....</b>		<b>67</b>
<b>ENCLOSURE A .....</b>		<b>A1</b>
A.1	<i>System's Overall Simulink</i> .....	A1
A.2	<i>Roll Angle Control Simulink</i> .....	A1
A.3	<i>Pitch Angle Control Simulink</i> .....	A1
A.4	<i>Yaw Angle Control Simulink</i> .....	A2
A.5	<i>X Position Control Simulink</i> .....	A2
A.6	<i>Y Position Control Simulink</i> .....	A2
A.7	<i>Z Position Control Simulink</i> .....	A2
A.8	<i>Switch Manual for Noise Simulink</i> .....	A3
A.10	<i>Rotational Linearization Simulink</i> .....	A4
A.11	<i>Translational Linearization Simulink</i> .....	A5
<b>ENCLOSURE B .....</b>		<b>B1</b>
B.1	<i>X Position's Adaptive PD Matlab Program</i> .....	B1
B.2	<i>Y Position's Adaptive PD Matlab Program</i> .....	B1
<b>BIOGRAPHY .....</b>		<b>C1</b>



## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Konfigurasi <i>Quadcopter</i> .....	8
<b>Gambar 2.2</b>	Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Thrust</i> .....	9
<b>Gambar 2.3</b>	Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Roll</i> .....	9
<b>Gambar 2.4</b>	Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Pitch</i> .....	10
<b>Gambar 2.5</b>	Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Yaw</i> .....	10
<b>Gambar 2.6</b>	<i>Earth-frame</i> dan <i>Body-frame</i> dari <i>Quadcopter</i> .....	11
<b>Gambar 2.7</b>	Rotasi <i>Quadcopter</i> Sumbu Z.....	12
<b>Gambar 2.8</b>	Rotasi <i>Quadcopter</i> Sumbu Y .....	13
<b>Gambar 2.9</b>	Rotasi <i>Quadcopter</i> Sumbu X .....	14
<b>Gambar 2.10</b>	Blok Diagram <i>Closed Loop</i> .....	26
<b>Gambar 2.11</b>	Skema Blok Diagram Rancangan Sistem.....	26
<b>Gambar 2.12</b>	Blok Diagram Kontroler Proporsional .....	27
<b>Gambar 2.13</b>	Blok Diagram Kontroler Derivatif .....	27
<b>Gambar 2.14</b>	Mode Tiruan Sebuah <i>Neuron</i> .....	28
<b>Gambar 2.15</b>	Skema <i>Back Propagation Learning</i> .....	29
<b>Gambar 2.16</b>	Skema <i>Back Propagation Learning</i> dari PD Adaptif .....	29
<b>Gambar 3.1</b>	Blok Diagram Pengendalian Sudut <i>Roll</i> .....	37
<b>Gambar 3.2</b>	Blok Diagram Pengendalian Sudut <i>Pitch</i> .....	39
<b>Gambar 3.3</b>	Skema Blok Diagram PD Adaptif .....	41
<b>Gambar 3.4</b>	Skema Jaringan Saraf Tiruan Kontroler PD Adaptif .....	39
<b>Gambar 4.1</b>	Skema Simulink Simulasi Pergerakan <i>Quadcopter</i> . .....	46
<b>Gambar 4.2</b>	Blok Diagram Simulink <i>Switch Manual</i> .....	46
<b>Gambar 4.3</b>	Stabilisasi Sudut <i>Roll</i> .....	47
<b>Gambar 4.4</b>	Stabilisasi Sudut <i>Pitch</i> .....	47
<b>Gambar 4.5</b>	Stabilisasi Sudut <i>Yaw</i> .....	48
<b>Gambar 4.6</b>	Respon Posisi Lintasan pada Sumbu-X tanpa <i>Noise</i> .....	49
<b>Gambar 4.7</b>	Respon Posisi Lintasan pada Sumbu-Y tanpa <i>Noise</i> .....	50
<b>Gambar 4.8</b>	Respon Posisi Lintasan pada Sumbu-Z tanpa <i>Noise</i> .....	52

<b>Gambar 4.9</b>	Posisi <i>Trajectory Tracking</i> tanpa <i>Noise</i> .....	52
<b>Gambar 4.10</b>	Sinyal U2 saat Dilakukan Simulasi tanpa <i>Noise</i> .....	53
<b>Gambar 4.11</b>	Sinyal U3 saat Dilakukan Simulasi tanpa <i>Noise</i> .....	54
<b>Gambar 4.12</b>	Respon Posisi Lintasan pada Sumbu-X dengan <i>Noise</i> Konstan .....	55
<b>Gambar 4.13</b>	Respon Posisi Lintasan pada Sumbu-Y dengan <i>Noise</i> Konstan .....	57
<b>Gambar 4.14</b>	Respon Posisi Lintasan pada Sumbu-Z dengan <i>Noise</i> Konstan .....	58
<b>Gambar 4.15</b>	Posisi <i>Trajectory Tracking</i> dengan <i>Noise</i> Konstan .	59
<b>Gambar 4.16</b>	Respon Posisi Lintasan pada Sumbu-X dengan <i>Random</i> <i>Noise</i> .....	60
<b>Gambar 4.17</b>	Respon Posisi Lintasan pada Sumbu-Y dengan <i>Random</i> <i>Noise</i> .....	62
<b>Gambar 4.18</b>	Respon Posisi Lintasan pada Sumbu-Z dengan <i>Random</i> <i>Noise</i> .....	63
<b>Gambar 4.12</b>	Posisi <i>Trajectory Tracking</i> dengan <i>Random Noise</i> .	64

## DAFTAR TABEL

<b>Tabel 3.1</b>	Parameter <i>Quadcopter</i> .....	33
<b>Tabel 3.2</b>	Nilai Kp dan Kd Kontroler PD Sudut <i>Roll</i> .....	38
<b>Tabel 3.3</b>	Nilai Kp dan Kd Kontroler PD Sudut <i>Pitch</i> .....	39
<b>Tabel 3.4</b>	Nilai Kp dan Kd Kontroler PD Sudut <i>Yaw</i> .....	40
<b>Tabel 3.5</b>	Nilai Kp dan Kd Sontroler PD Gerak <i>Heve</i> pada Sumbu-Z.....	40
<b>Tabel 3.6</b>	Parameter Kontroler PD Adaptif Gerak Translasi pada Sumbu-X.....	42
<b>Tabel 3.7</b>	Parameter Kontroler PD Adaptif Gerak Translasi pada Sumbu-Y.....	42

*--halaman ini sengaja dikosongkan--*

# BAB I

## PENDAHULUAN

Tugas Akhir adalah suatu penelitian yang bersifat mandiri yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Topik yang dibahas dalam Tugas Akhir ini ialah mengenai *trajectory tracking* gerak *cruise* pada *quadcopter*.

Pada Bab ini membahas mengenai hal-hal yang mendahului pelaksanaan Tugas Akhir. Hal tersebut meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

### 1.1 Latar Belakang

Beberapa tahun terakhir, teknologi penerbangan berkembang dengan pesat. Teknologi tersebut dituntut untuk dapat membantu kinerja manusia. Salah satu contohnya ialah kendaraan udara tak berawak atau UAV (*Unmanned Aerial Vehicle*). UAV memiliki banyak kegunaan dalam berbagai bidang khususnya pada bidang militer dan sipil. UAV memiliki berbagai jenis, yaitu *fixed wing*, *axial wing*, *coaxial wing*, dan *Rotary wing*.

Salah satu jenis *Rotary wing* adalah *Quadcopter*. *Quadcopter* sering disebut *Quadcopter* karena memiliki 4 *propeller* dengan konfigurasi menyilang. *Propeller* bagian depan dan belakang akan berputar searah dengan jarum jam (*clockwise*). Bagian kiri dan kanan akan berputar berlawanan arah jarum jam (*counter clockwise*).

Pergerakan *quadcopter* biasanya dikendalikan oleh *user* menggunakan *remote control*. Bila ingin menuju ke suatu tempat harus dikendalikan lewat *remote control*. Penggunaan alat ini biasanya tidak terlalu akurat bila menuju ke suatu tempat. Oleh karena itu untuk membuat pergerakan *quadcopter* menjadi lebih akurat, maka digunakan *Trajectory Tracking*. Dengan menggunakan *trajectory tracking* diharapkan *quadcopter* mampu menuju lokasi tujuan dalam waktu yang diinginkan. *Tracking* yang dilakukan *quadcopter* adalah *tracking* untuk gerakan *cruise*. Gerakan ini menuntut *quadcopter* untuk mampu melakukan maneuver untuk mencapai tempat tujuannya.

Pada tugas akhir ini, kontroler *tracking* akan di desain dengan menggunakan metode *adaptive compensation for position tracking error*

of *quadcopter UAVs*. Diharapkan dengan diterapkannya metode ini dapat mengurangi tingkat *error* yang terjadi saat melakukan *position tracking*.

Penerapan penelitian ini dapat digunakan pada *quadcopter* untuk melakukan navigasi, liputan, pemetaan wilayah, serta kegiatan lain yang membutuhkan *quadcopter* melakukan banyak gerakan manuver untuk menyelesaikannya.

## 1.2 Permasalahan

Permasalahan dalam perancangan serta pengujian Tugas Akhir ini adalah sebagai berikut:

1. Munculnya nilai *error* antara posisi referensi dan respon posisi saat *quadcopter* melakukan *trajectory tracking* gerak *cruise*.
2. Nilai *error* antara posisi referensi dan respon posisi saat *quadcopter* melakukan *trajectory tracking* gerak *cruise* membesar saat terkena gangguan atau *disturbance* berupa angin.

## 1.3 Batasan Permasalahan

Batasan masalah dalam perancangan serta pengujian Tugas Akhir ini adalah sebagai berikut:

1. Model dan parameter *Quadcopter* yang digunakan adalah model *Quadcopter* Quanser Q-Ball X-4.
2. Konfigurasi *propeller* dari *Quadcopter* berbentuk “plus”.
3. Simulasi penerbangan *Quadcopter* diasumsikan terbang didalam ruangan.
4. Titik origin *frame* inersia bumi dan *frame* badan *Quadcopter* berhimpit.
5. *Tracking* posisi dilakukan setelah *Quadcopter* terbang
6. Diasumsikan terdapat gangguan (*disturbances*) yang bernilai konstan saat pensimulasian gerak *tracking quadcopter*.
7. Sudut *Yaw* saat melakukan gerak lateral dianggap nol, karena gerakan yang dilakukan *quadcopter* tidak membutuhkan gerakan akrobatik yang signifikan.
8. Orientasi *quadcopter* (*heading*) berada pada sumbu-X negatif.

## 1.4 Tujuan

Tujuan dalam perancangan serta pengujian Tugas Akhir ini adalah sebagai berikut:

1. *Quadcopter* mampu melakukan *trajectory tracking* sesuai dengan rute yang diinginkan, dengan nilai  $error \leq 5\%$  dari posisi referensi yang diberikan.
2. *Quadcopter* mampu melakukan *trajectory tracking* saat mendapatkan disturbances sesuai dengan rute yang diinginkan, dengan nilai  $error \leq 5\%$  dari posisi referensi yang diberikan.

## 1.5 Metodologi

Dalam penelitian Tugas Akhir ini diperlukan suatu tahapan yang merepresentasikan urutan yang harus dilaksanakan agar sesuai dengan tujuan penelitian. Tahapan tersebut ialah sebagai berikut:

1. Studi literatur

Pada tahap ini akan dilakukan kegiatan pengumpulan dan pengkajian hal-hal terkait teori, informasi maupun hasil eksperimen serupa yang dapat dijadikan referensi dalam proses penyusunan tugas akhir ini. Sumber yang dikumpulkan dan dikaji dapat diperoleh melalui berbagai sumber ilmiah seperti diktat, buku, hasil penelitian, maupun jurnal ilmiah yang telah dipublikasikan. Studi literatur difokuskan mengenai konfigurasi dan cara kerja *quadcopter*, mempelajari model matematika *quadcopter*, cara mapping terrain, *trajectory tracking* dan implementasi dari kontroler yang telah dibuat.

2. Identifikasi

Tahap yang kedua ialah mendapatkan model matematika dari *quadcopter*. Model matematika tersebut didapatkan dengan menggunakan identifikasi parametric, yang telah dipelajari melalui studi literatur. Dengan diduplikasinya model matematika dari *quadcopter*, perancangan kontroler dapat dilakukan.

3. Perancangan Kontroler

Pada tahap ini, telah didapatkan lintasan yang akan dilalui *quadcopter*, sehingga untuk memastikan *quadcopter* dapat melintasi lintasan dengan baik, maka dirancang kontroler. Pada *quadcopter* ini digunakan kontroler PD untuk rotasi dan translasi, serta kontroler Adaptif untuk meminimalisir terjadinya *error* saat *trajectory tracking*. Perancangan kontroler ini menggunakan software Matlab R2015a.

4. Simulasi dan Evaluasi

Tahap yang keempat ialah kontroler yang telah dirancang menggunakan *software* Matlab R2015a disimulasikan. Tujuan dari simulasi ialah untuk mengetahui performansi dari sistem yang telah diberi kontroler. Jika performansi yang didapatkan tidak sesuai dengan yang diinginkan maka dilakukan evaluasi. Evaluasi dilakukan dengan mengubah parameter dari kontroler yang telah dirancang.

5. Penulisan buku Tugas Akhir

Tahap yang terakhir ialah penulisan laporan/buku Tugas Akhir. Penulisan dilakukan secara intensif bila proses pengujian telah selesai.

## 1.6 Sistematika Penulisan

Tahap terakhir dari sebuah penelitian adalah penulisan laporan. Pada penulisan laporan/buku Tugas Akhir ini disusun berdasarkan 5 bab, di mana setiap bab berisi mengenai permasalahan dalam penelitian. Bab tersebut ialah sebagai berikut:

### **BAB I PENDAHULUAN**

Membahas latar belakang, permasalahan, pembatasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi pembahasan tugas akhir ini.

### **BAB II TEORI PENUNJANG**

Membahas konsep dasar dan teori yang mendasari perancangan tugas akhir ini, meliputi teori *quadcopter*, kinematika dan dinamika *quadcopter*, konsep dasar kontroler PD dan Adaptif, dan konsep dasar linierisasi dengan metode *feedback linearization*.

### **BAB III PERANCANGAN SISTEM**

Membahas model dinamika *quadcopter*, linierisasi, perancangan kontroler *quadcopter* dengan mengacu teori pada BAB II.

### **BAB IV PENGUJIAN DAN ANALISIS**

Membahas prosedur pelaksanaan pengujian dan analisis data mengenai pengendalian gerak *cruise* menggunakan kontroler PD dan Adaptif.

### **BAB V PENUTUP**



Berisi mengenai kesimpulan dari penelitian Tugas Akhir dan saran untuk dapat digunakan untuk pengembangan tugas akhir ini untuk lebih lanjut.

## **1.7 Relevansi**

Relevansi dalam perancangan serta pengujian Tugas Akhir ini adalah sebagai berikut:

1. Pengembangan *quadcopter* khususnya saat melakukan gerak *cruise*.
2. Perbandingan metode kontroler pada *quadcopter* saat melakukan gerak *cruise*.

*--halaman ini sengaja dikosongkan--*

## **BAB II**

### **TEORI PENUNJANG**

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat beberapa teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian awal terdapat tinjauan pustaka yang menggambarkan landasan teori mengenai *quadcopter* secara umum yang akan digunakan pada Tugas Akhir ini. Pada bagian selanjutnya membahas mengenai teori-teori pendukung, meliputi dinamika dan kinematika *quadcopter*, kontroler PD dan kontroler PD Adaptif.

#### **2.1 *Quadcopter* [1]**

*Quadcopter* merupakan suatu robot terbang kecil UAV (*Unmanned Aerial Vehicle*) dengan model mekanik yang terdiri atas empat buah rotor sebagai penggeraknya. Masing-masing rotor tersebut menghasilkan daya angkat yang dan memiliki jarak yang sama menuju titik pusat massa *quadcopter*. Dengan daya angkat dari keempat rotor tersebut, maka *quadcopter* dapat terbang. Adapun kecepatan *quadcopter* bergantung kepada kekuatan dan motor dan berat *quadcopter* itu sendiri.

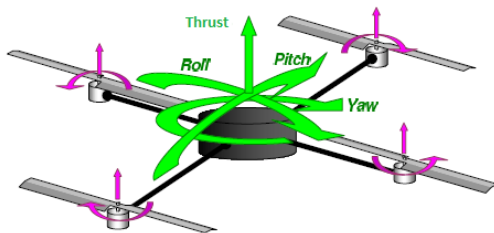
*Quadcopter* mampu melakukan gerak *take-off* dan *landing* secara longitudinal atau istilah lainnya vertikal. *Quadcopter* juga mampu melakukan gerakan-gerakan *maneuver* yang tidak dapat dilakukan oleh UAV lain, seperti gerakan *heve* keatas dan kebawah, gerakan *surf* kedepan dan kebelakang, serta gerak *sway* kekiri dan kenanan. Gerakan-gerakan ini dilakukan *quadcopter* dengan cara mengubah kecepatan putaran tiap-tiap motornya.

##### **2.1.1 Konsep Dasar *Quadcopter* [1], [2]**

*Quadcopter* yang digunakan sebagai robot terbang kecil memiliki model mekanik yang terdiri dari empat *rotor* yang dipasang pada sumbu *plus* (+) simetris. Bentuk ini diharapkan tipis dan kaku, sehingga diperoleh friksi udara yang kecil dan komponen yang bergerak pada *quadcopter* hanyalah putaran *propeller*. Setiap *propeller* pada *quadcopter*

diputar oleh satu motor elektrik, sehingga terdapat empat motor sebagai aktuator untuk menghasilkan gaya angkat dari *quadcopter*.

Dengan batasan menggunakan karakteristik motor dan *propeller* yang relatif sama, maka kondisi melayang (*hover*) akan diperoleh kecepatan motor yang sama disetiap *propeller*. Konfigurasi *propeller* terdiri dari dua macam, yaitu dua *propeller* yang bergerak searah jarum jam dan dua *propeller* yang bergerak berlawanan arah jarum jam di mana setiap satu sumbu, *propeller* berputar dengan arah yang sama yang terdapat pada Gambar 2.1.

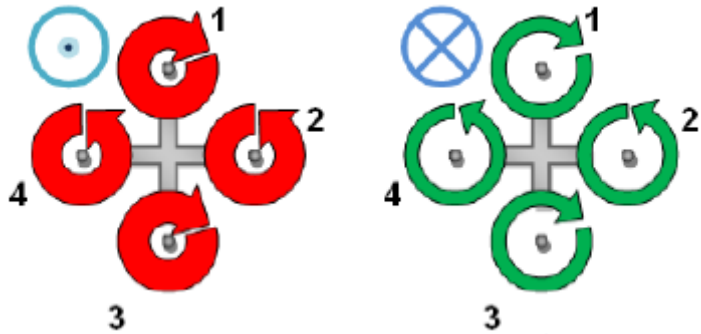


**Gambar 2.1** Konfigurasi *Quadcopter*

Dengan melakukan pengaturan kecepatan putaran *propeller* akan dihasilkan beberapa komando *input*, diantaranya ialah:

a. Gaya *Thrust* ( $U_1$ )

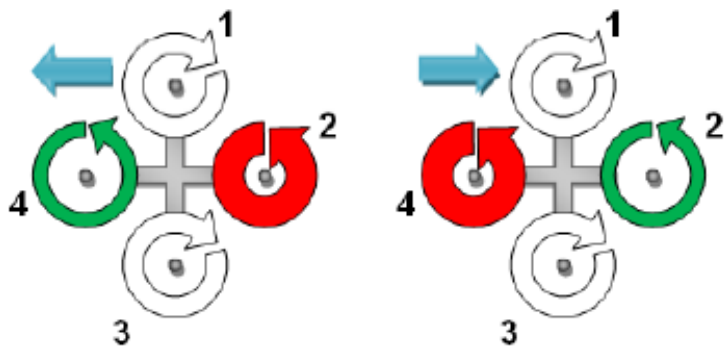
Dengan mempercepat (warna merah) atau memperlambat (warna hijau) kecepatan motor dari propeler secara bersamaan akan menghasilkan percepatan vertikal yang ditunjukkan pada Gambar 2.2.



**Gambar 2.2** Konfigurasi Arah Putaran untuk Membangkitkan Gaya *Thrust*

b. Torsi *Roll* ( $U_2$ )

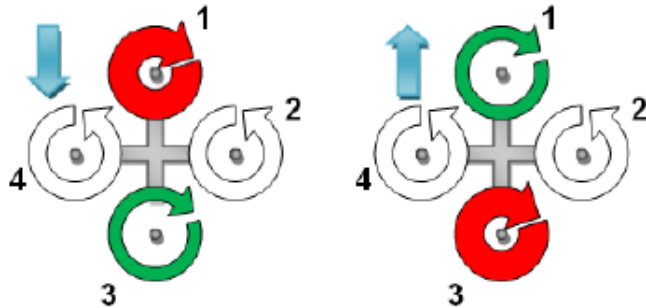
Dengan mempercepat atau memperlambat propeler pada sisi kiri dan secara bersamaan memperlambat atau mempercepat propeler di sisi kanan akan diperoleh rotasi dengan percepatan sudut yang ditunjukkan pada Gambar 2.3.



**Gambar 2.3** Konfigurasi Arah Putaran untuk Membangkitkan Gaya *Roll*

c. Torsi *Pitch* ( $U_3$ )

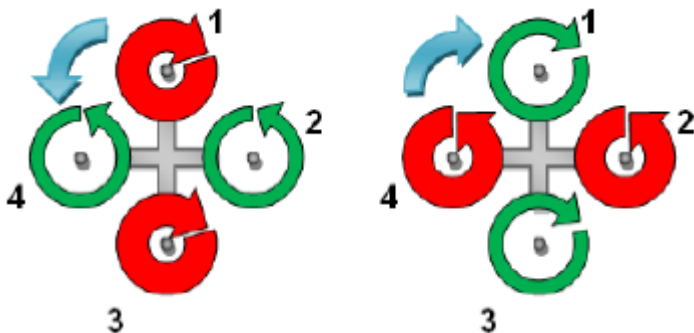
Pada Gambar 2.4 mekanisme yang sama seperti gerak *roll*, namun dilakukan pada propeler di sisi depan dan belakang akan diperoleh percepatan sudut *pitch*.



**Gambar 2.4** Konfigurasi Arah Putaran untuk Membangkitkan Gaya *Pitch*

d. Torsi *Yaw* ( $U_4$ )

Dengan mempercepat atau memperlambat propeler yang berputar berlawanan arah jarum jam, dan memperlambat atau mempercepat propeler yang berputar searah jarum jam akan diperoleh gerak *yaw* dengan percepatan sudut. Gerakan ini berputar dengan acuan pada sumbu Z. Gerakan *yaw* ditunjukkan pada Gambar 2.5.



**Gambar 2.5** Konfigurasi Arah Putaran untuk Membangkitkan Gaya *Yaw*

### 2.1.2 Kinematika *Quadcopter* [2], [3]

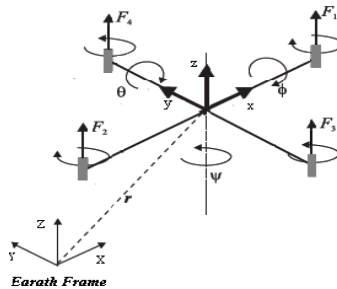
Sebelum melakukan analisis kinematika, dilakukan beberapa penetapan-penetapan yang akan digunakan. Pertama, analisis yang digunakan untuk kinematika memakai bingkai diagram cartesius tiga dimensi (X, Y, Z). Bingkai diagram cartesius dibagi menjadi dua, yaitu bingkai bumi (*earth frame*) yang kaku atau tidak bergerak, serta bingkai *body quadcopter* yang bergerak rotasi dan translasi. Ditetapkan sumbu X-bumi berada pada garis utara-selatan, sumbu Y-bumi berada pada garis timur-barat, sumbu Z-bumi berada pada garis vertikal menuju pusat bumi, sumbu X-*quadcopter* berada pada garis depan-belakang *quadcopter*, sumbu Y-*quadcopter* berada pada garis kiri-kanan *quadcopter*, dan sumbu Z-*quadcopter* berada pada garis atas-bawah *quadcopter*.

Bila didefinisikan sumbu  $X^+$  menuju utara dan depan *quadcopter*, sumbu  $Z^+$  menjauhi pusat bumi dan menuju atas *quadcopter* maka berlandaskan aturan kaidah tangan kanan *cross vector* dapat ditentukan bahwa sumbu  $Y^+$  menuju barat atau kiri *quadcopter* seperti yang ditampilkan pada Gambar 2.6.

Posisi linear *quadcopter* ( $\Gamma^E$ ) ditentukan dari koordinat vektor antara origin *B-frame* serta origin dari *E-frame* dengan memperhatikan *E-frame*. Posisi angular *quadcopter* ( $\Theta^E$ ) ditentukan dari orientasi *B-frame* terhadap *E-frame*. Posisi linear dan posisi angular terdapat pada Persamaan (2.1)-(2.2).

$$\Gamma^E = [X \ Y \ Z]^T \quad (2.1)$$

$$\Theta^E = [\phi \ \theta \ \psi]^T \quad (2.2)$$

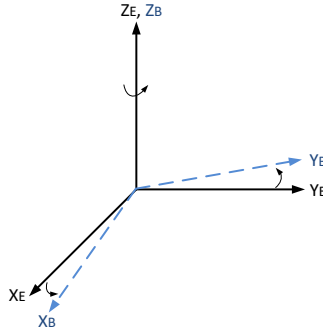


**Gambar 2.6** Earth-frame dan Body-frame dari *Quadcopter*

Matriks rotasi dari *quadcopter* diperoleh dengan mengkalikan matriks transformasi untuk rotasi di tiap sumbu.

a. Rotasi Sumbu Z

*Quadcopter* berotasi pada sumbu Z dan menghasilkan sudut *yaw* yang dilambangkan dengan  $R(\psi, z)$ . Rotasi sudut *yaw* terdapat pada Gambar 2.7



**Gambar 2.7** Rotasi *Quadcopter* Sumbu Z

$$\begin{cases} X_E = X_B \cos \psi - Y_B \sin \psi \\ Y_E = X_B \sin \psi + Y_B \cos \psi \\ Z_E = Z_B \end{cases} \quad (2.3)$$

Persamaan (2.3) dapat dibuat dalam bentuk matriks pada Persamaan (2.4).

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.4)$$

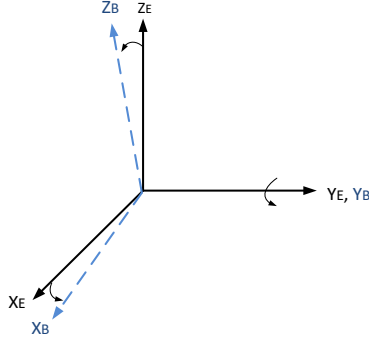
Sehingga didapatkan matriks rotasi pada sumbu X ditunjukkan pada Persamaan (2.5).

$$R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$



b. Rotasi Sumbu Y

*Quadcopter* berotasi pada sumbu Y dan menghasilkan sudut *pitch* yang dilambangkan dengan  $R(\theta, y)$ . Rotasi sudut *pitch* terdapat pada Gambar 2.8



**Gambar 2.8** Rotasi *Quadcopter* Sumbu Y

$$\begin{cases} X_E = X_B \cos \theta + Z_B \sin \theta \\ Y_E = Y_B \\ Z_E = -X_B \sin \theta + Z_B \cos \theta \end{cases} \quad (2.6)$$

Persamaan (2.6) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.7).

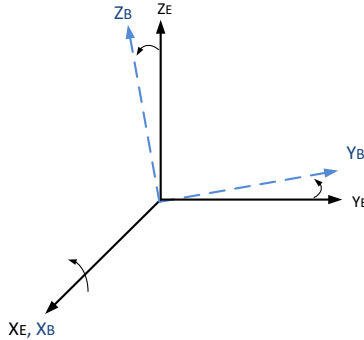
$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.7)$$

Adapun matriks rotasi pada sumbu Y ditunjukkan pada Persamaan (2.8).

$$R(\theta, y) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.8)$$

c. Rotasi Sumbu X

*Quadcopter* berotasi pada sumbu X dan menghasilkan sudut *roll* yang dilambangkan dengan  $R(\phi, x)$ . Rotasi sudut *roll* terdapat pada Gambar 2.7



**Gambar 2.9** Rotasi *Quadcopter* Sumbu X

$$\begin{cases} X_E = X_B \\ Y_E = Y_B \cos \phi - Y_B \sin \phi \\ Z_E = Y_B \sin \phi + Z_B \cos \phi \end{cases} \quad (2.9)$$

Persamaan (2.9) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.10).

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.10)$$

Adapun matriks rotasi pada sumbu Z ditunjukkan pada Persamaan (2.11).

$$R(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.11)$$

Sehingga persamaan rotasi *quadcopter* terdapat pada Persamaan (2.12)-(2.13).

$$R_{\theta} = R(\phi)R(\theta)R(\psi) \quad (2.12)$$

$$R_{\theta} = \begin{bmatrix} c_{\psi}c_{\theta} & -s_{\psi}c_{\phi} + c_{\psi}s_{\theta}s_{\phi} & s_{\psi}s_{\phi} + c_{\psi}s_{\theta}s_{\phi} \\ s_{\psi}c_{\theta} & c_{\psi}c_{\phi} + s_{\psi}s_{\theta}s_{\phi} & -c_{\psi}s_{\phi} + s_{\psi}s_{\theta}s_{\phi} \\ -s_{\theta} & c_{\theta}s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix} \quad (2.13)$$

Sedangkan untuk kecepatan dalam *quadcopter* diekspresikan terhadap *body frame* (*B-frame*). Kecepatan *quadcopter* terdiri dari kecepatan linear  $V^B$  dan kecepatan angular  $\omega^B$ . Komposisi vektornya disajikan dalam Persamaan (2.14).

$$\omega^B = [p \quad q \quad r]^T \quad (2.14)$$

Diperlukan kombinasi nilai linear dan angular untuk memberikan representasi yang lengkap dalam *space*.  $\xi$  merupakan komposisi dari vektor posisi linear  $\Gamma^E$  (m) dan vektor posisi sudut  $\Theta^E$  (rad) terhadap *earth frame* (*E-frame*) seperti terlihat pada Persamaan (2.15).

$$\xi = [\Gamma^E \quad \Theta^E]^T = [X \quad Y \quad Z \quad \phi \quad \theta \quad \psi]^T \quad (2.15)$$

$V$  merupakan generalisasi dari vektor kecepatan linear *quadcopter*  $V^B$  ( $\text{m s}^{-1}$ ) dan kecepatan angular *quadcopter*  $\omega^B$  ( $\text{rad s}^{-1}$ ) pada *body frame* (*B-frame*) yang terdapat pada Persamaan (2.17).

$$V = [V^B \quad \omega^B]^T = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (2.16)$$

Hubungan antara kecepatan linear pada *body frame* (*B-frame*) dan salah satu factor pada *earth frame* (*E-frame*)  $V^E$  (atau  $\dot{\Gamma}^E$ ) [ $\text{m/s}$ ] dapat dilihat pada Persamaan (2.17).

$$V^E = \dot{\Gamma}^E = R_{\theta} V^B \quad (2.17)$$

Dimana  $R_{\theta}$  adalah matrik rotasi dari *body frame* (*B-frame*) ke *earth frame* (*E-frame*). Seperti pada kecepatan linear, hal tersebut juga berlaku untuk menghubungkan kecepatan angular pada *E-frame* (atau

kecepatan Euler)  $\dot{\theta}^E$  [rad/s] ke *B-frame*  $\omega^B$  atau sebaliknya. Hubungan tersebut terdapat pada Persamaan (2.18).

$$\omega^B = T_{\Theta}^{-1} \dot{\theta}^E \quad (2.18)$$

$$\dot{\theta}^E = T_{\Theta} \omega^B \quad (2.19)$$

Dimana  $T_{\Theta}$  adalah matriks transformasi. Matriks transformasi  $T_{\Theta}$  dapat ditetapkan dengan menggunakan kecepatan Euler dalam *B-frame*, dengan membalik pola perputaran sudut dari *roll*, *pitch* dan *yaw* seperti pada Persamaan (2.20).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)^{-1} R(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.20)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_{\Theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

Persamaan (2.13)-(2.21) maka diperoleh matriks transformasi dari *body frame* (*B-frame*) menuju *earth frame* (*E-frame*).

$$T_{\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta} s_{\phi} \\ 0 & -s_{\phi} & c_{\phi} c_{\theta} \end{bmatrix} \quad (2.22)$$

$$T_{\Theta} = \begin{bmatrix} 1 & s_{\phi} t_{\theta} & c_{\phi} t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi} / c_{\theta} & c_{\phi} / c_{\theta} \end{bmatrix} \quad (2.23)$$

Persamaan yang sudah didapat, maka dibentuk suatu hubungan antara kecepatan terhadap *earth frame* (*E-frame*) dan *body frame* (*B-frame*).

$$\dot{\xi} = J_{\Theta} v \quad (2.24)$$

Dimana  $\dot{\xi}$  adalah vektor kecepatan yang mengacu pada *earth frame* (*E-frame*),  $v$  adalah vektor kecepatan mengacu *body frame* (*B-frame*) dan  $J_{\Theta}$  adalah matrik *jacobian*. Matrik *jacobian* terdiri dari 4 sub-matrik sebagaimana Persamaan (2.25).

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta} \end{bmatrix} \quad (2.25)$$

### 2.1.3 Dinamika *Quadcopter* [2], [3]

Berdasarkan aksioma pertama Euler dari hukum kedua Newton, turunan dari komponen linear dari gerakan suatu benda dapat dilihat pada Persamaan (2.26).

$$\left\{ \begin{array}{l} m \ddot{\Gamma}^E = F^E \\ m \overbrace{\dot{R}_{\Theta} V^B} = R_{\Theta} F^B \\ m (R_{\Theta} \dot{V}^B + \dot{R}_{\Theta} V^B) = F^B \\ m R_{\Theta} (\dot{V}^B + \omega^B \times V^B) = R_{\Theta} F^B \\ m (\dot{V}^B + \omega^B \times V^B) = F^B \end{array} \right. \quad (2.26)$$

Dengan  $m$  (kg) adalah massa *quadcopter*,  $\ddot{\Gamma}^E$  ( $m s^{-2}$ ) adalah vektor percepatan linear *quadcopter* yang mengacu pada E-frame,  $F^E$  (N) adalah vektor gaya *quadcopter* terhadap E-frame,  $\dot{V}^B$  ( $m s^{-2}$ ) adalah percepatan linear *quadcopter* terhadap B-frame, dan  $\dot{R}_{\Theta}$  adalah turunan pertama matriks rotasi. Dan simbol  $\times$  merupakan perkalian produk suatu vektor.

Berdasarkan aksioma kedua Euler dari hukum kedua Newton, dengan cara yang sama, turunan dari gerakan angular komponen dari suatu benda dapat dilihat pada Persamaan (2.27).

$$\left\{ \begin{array}{l} I \ddot{\Theta}^E = \tau^E \\ I \overbrace{\dot{T}_{\Theta} \omega^B} = T_{\Theta} \tau^B \\ I \dot{\omega}^B + \omega^B \times (I \omega^B) = \tau^B \end{array} \right. \quad (2.27)$$

Pada Persamaan (2.27),  $I$  ( $N m s^2$ ) adalah matriks inersia *quadcopter* (pada B-frame),  $\ddot{\Theta}^E$  ( $rad s^{-2}$ ) adalah vektor percepatan sudut *quadcopter* terhadap E-frame,  $\dot{\omega}^B$  ( $rad s^{-2}$ ) adalah vektor percepatan sudut *quadcopter* terhadap B-frame, dan  $\tau^E$  (N m) adalah torsi *quadcopter* terhadap E-frame.

Dari Persamaan (2.26) dan (2.27), dapat dibentuk persamaan benda kaku dengan 6 derajat kebebasan (DOF). Persamaan (2.28) menunjukkan formulasi matriks dari dinamika sistem.

$$\begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}^B \\ \boldsymbol{\omega}^B \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^B \times (m \mathbf{V}^B) \\ \boldsymbol{\omega}^B \times (I \boldsymbol{\omega}^B) \end{bmatrix} = \begin{bmatrix} \mathbf{F}^B \\ \boldsymbol{\tau}^B \end{bmatrix} \quad (2.28)$$

Dimana notasi  $I_{3 \times 3}$  adalah matriks identitas  $3 \times 3$ . Dan  $0_{3 \times 3}$  adalah matriks nol  $3 \times 3$ . *quadcopter* memiliki empat buah motor yang menghasilkan gaya dorong seperti pada Gambar 2.6.

Vektor gaya yang terjadi pada *quadcopter* dapat ditentukan berdasarkan Persamaan (2.29).

$$\boldsymbol{\Lambda} = [\mathbf{F}^B \ \boldsymbol{\tau}^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \quad (2.29)$$

Persamaan (2.29) dapat juga ditulis dalam bentuk formulasi matriks seperti pada Persamaan (2.30).

$$\mathbf{M}_B \dot{\mathbf{v}} + \mathbf{C}_B(\mathbf{v})\mathbf{v} = \boldsymbol{\Lambda} \quad (2.30)$$

Dengan  $\dot{\mathbf{v}}$  adalah vektor percepatan *quadcopter* terhadap B-frame.  $\mathbf{M}_B$  adalah matriks inersia sistem dan  $\mathbf{C}_B$  adalah matriks sentripetal Coriolis. Persamaan (2.31) menunjukkan matriks  $\mathbf{M}_B$ .

$$\mathbf{M}_B = \begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (2.31)$$

Dapat dilihat bahwa matriks  $\mathbf{M}_B$  adalah matriks diagonal, sedangkan matriks  $\mathbf{C}_B$  ditunjukkan oleh Persamaan (2.32).

$$\mathbf{C}_B = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -m \mathbf{S}(\mathbf{V}_B) \\ \mathbf{0}_{3 \times 3} & -\mathbf{S}(I \boldsymbol{\omega}_B) \end{bmatrix}$$

$$\mathbf{C}_B = \begin{bmatrix} 0 & 0 & 0 & 0 & m w & -m v \\ 0 & 0 & 0 & -m w & 0 & m u \\ 0 & 0 & 0 & m v & -m u & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ} r & -I_{YY} q \\ 0 & 0 & 0 & -I_{ZZ} r & 0 & I_{XX} p \\ 0 & 0 & 0 & I_{YY} q & -I_{XX} p & 0 \end{bmatrix} \quad (2.32)$$

Pada persamaan ini, diadopsi operator *skew-symmetric*  $\mathbf{S}(\cdot)$ . Untuk vektor 3 dimensi  $\mathbf{k}$ , matriks *skew-symmetric* dari  $\mathbf{k}$  ( $\mathbf{S}(\mathbf{k})$ ) ditunjukkan pada Persamaan (2.33).

$$\mathbf{S}(\mathbf{k}) = -\mathbf{S}^T(\mathbf{k}) = \begin{bmatrix} 0 & -k_3 & k_1 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}, \mathbf{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (2.33)$$

Matriks  $\mathbf{A}$  dapat dibagi menjadi tiga komponen menurut kontribusi sifat dasar *quadcopter*. Percepatan gravitasi hanya berlaku pada persamaan linear *quadcopter*. Persamaan matriks  $\mathbf{G}_B(\xi)$  dituliskan pada Persamaan (2.34).

$$\begin{aligned} \mathbf{G}_B(\xi) &= \begin{bmatrix} \mathbf{F}_{GB} \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\Theta^{-1} \mathbf{F}_{GE} \\ 0_{3 \times 1} \end{bmatrix} \\ \mathbf{G}_B(\xi) &= \begin{bmatrix} \mathbf{R}_\Theta^T & \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} mg \sin \theta \\ -mg \cos \theta \sin \phi \\ -mg \cos \theta \sin \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (2.34)$$

di mana  $\mathbf{F}_{GB}$  (N) adalah vektor gaya gravitasi terhadap B-frame dan  $\mathbf{F}_{GE}$  (N) terhadap E-frame.  $\mathbf{R}_\Theta$  adalah matriks ortogonal, sehingga matriks inversnya sama dengan matriks transposnya.

Kontribusi yang kedua mengenai efek giroskopis yang dihasilkan oleh perputaran *propeller*. Selama dua *propeller* yang berhadapan berputar searah jarum jam dan dua *propeller* lainnya berputar berlawanan arah jarum jam, terjadi ketidak-seimbangan saat jumlah aljabar dari kecepatan motor tidak sama dengan nol. Jika *roll* dan *pitch* juga tidak nol, *quadcopter* akan mengalami torsi efek giroskopis yang ditunjukkan pada Persamaan (2.35).

$$\begin{aligned} \mathbf{O}_B(\mathbf{v})\boldsymbol{\Omega} &= \left[ -\sum_{k=1}^4 J_{TP} \left( \begin{matrix} 0_{3 \times 1} \\ \boldsymbol{\omega}^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{matrix} \right) (-1)^k \boldsymbol{\Omega}_k \right] = \left[ J_{TP} \begin{bmatrix} 0_{3 \times 1} \\ -q \\ p \\ 0 \end{bmatrix} \boldsymbol{\Omega} \right] \\ \mathbf{O}_B(\mathbf{v})\boldsymbol{\Omega} &= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -q & -q & -q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega} \end{aligned} \quad (2.35)$$

$\mathbf{O}_B$  adalah matriks giroskopis dari *propeller* dan  $J_{TP}$  (N m s<sup>2</sup>) adalah momen inersia total di sekitar sumbu *propeller*. Efek giroskopis yang dihasilkan oleh putaran *propeller* hanya berhubungan dengan persamaan angular, dan tidak mempengaruhi persamaan linear.  $\boldsymbol{\Omega}$  (rad s<sup>-1</sup>) merupakan vektor kecepatan putar *propeller*, yang ditunjukkan oleh Persamaan (2.36).

$$\boldsymbol{\Omega} = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \quad \boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (2.36)$$

di mana  $\Omega_1$ ,  $\Omega_2$ ,  $\Omega_3$ , dan  $\Omega_4$  (rad s<sup>-1</sup>) adalah kecepatan *propeller* depan, kanan, belakang, dan kiri.

Kontribusi yang ketiga adalah perhitungan gaya dan torsi yang dihasilkan oleh pergerakan *input*. Pergerakan matriks  $\mathbf{E}_B$  dikali dengan  $\boldsymbol{\Omega}^2$  untuk memperoleh vektor perpindahan  $\mathbf{U}_B(\boldsymbol{\Omega})$ . Efek aerodinamis (factor *thrust*  $b$  (N s<sup>2</sup>) dan *drag*  $d$  (N m s<sup>2</sup>)) berpengaruh pada gaya dan torsi yang dihasilkan. Persamaan vektor perpindahan pada dinamika *quadcopter* ditunjukkan pada Persamaan (2.37).

$$\mathbf{U}_B(\boldsymbol{\Omega}) = \mathbf{E}_B \boldsymbol{\Omega}^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$



$$\mathbf{U}_B(\boldsymbol{\Omega}) = \begin{bmatrix} 0 \\ 0 \\ b(\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_2^2 + \boldsymbol{\Omega}_3^2 + \boldsymbol{\Omega}_4^2) \\ bl(-\boldsymbol{\Omega}_2^2 + \boldsymbol{\Omega}_4^2) \\ bl(-\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_3^2) \\ d(-\boldsymbol{\Omega}_1^2 + \boldsymbol{\Omega}_2^2 - \boldsymbol{\Omega}_3^2 + \boldsymbol{\Omega}_4^2) \end{bmatrix} \quad (2.37)$$

di mana  $l$  (m) adalah jarak antara pusat *quadcopter* dengan pusat *propeller*.  $U_1, U_2, U_3$ , dan  $U_4$  adalah komponen vektor gerak. Hubungan komponen vektor tersebut dengan *propeller* dapat diperoleh dari perhitungan aerodinamis.

Dari persamaan sebelumnya, dapat diketahui konstanta matriks  $\mathbf{E}_B$  yang dikalikan dengan kuadrat dari kecepatan *propeller*  $\boldsymbol{\Omega}^2$  sehingga menghasilkan  $\mathbf{U}_B(\boldsymbol{\Omega})$ . Persamaan (2.38) menunjukkan matriks  $\mathbf{E}_B$ .

$$\mathbf{E}_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \quad (2.38)$$

Dari Persamaan (2.29), dinamika *quadcopter* dapat dibuat dengan mempertimbangkan ketiga kontribusi menurut Persamaan (2.39).

$$\mathbf{M}_B \dot{\mathbf{v}} + \mathbf{C}_B(\mathbf{v})\mathbf{v} = \mathbf{G}_B(\boldsymbol{\xi}) + \mathbf{O}_B(\mathbf{v}) \boldsymbol{\Omega} + \mathbf{E}_B \boldsymbol{\Omega}^2 \quad (2.39)$$

Dengan mengatur ulang Persamaan (2.39) diatas, turunan dari vektor kecepatan terhadap *B-frame* dapat dirumuskan sebagai berikut:

$$\dot{\mathbf{v}} = \mathbf{M}_B^{-1} (\mathbf{C}_B(\mathbf{v})\mathbf{v} + \mathbf{G}_B(\boldsymbol{\xi}) + \mathbf{O}_B(\mathbf{v}) \boldsymbol{\Omega} + \mathbf{E}_B \boldsymbol{\Omega}^2) \quad (2.40)$$

Persamaan (2.40) menunjukkan persamaan-persamaan sebelumnya bukan dalam bentuk matriks, melainkan dalam bentuk sistem Persamaan (2.41).

$$\begin{aligned}
\dot{u} &= (v r - w q) + g s_\theta \\
\dot{v} &= (w p - u r) - g c_\theta s_\phi \\
\dot{w} &= (u q - v p) - g c_\theta s_\phi + \frac{U_1}{m} \\
\dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}} q r - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}} \\
\dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}} p r - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_3}{I_{YY}} \\
\dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}} p q - \frac{U_4}{I_{ZZ}}
\end{aligned} \tag{2.41}$$

Persamaan *input* kecepatan *propeller* (baling-baling) dapat dilihat dalam Persamaan (2.42).

$$\begin{aligned}
U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
U_2 &= bl(-\Omega_2^2 + \Omega_4^2) \\
U_3 &= bl(-\Omega_1^2 + \Omega_3^2) \\
U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\
\Omega &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4
\end{aligned} \tag{2.42}$$

Persamaan sistem dinamik *quadcopter* pada Persamaan (2.41) diperoleh dari B-frame. Dalam pemodelannya, dibutuhkan persamaan dinamik sistem yang mencakup sistem *hybrid* yang disusun oleh persamaan linear dari E-frame dan persamaan angular dari B-frame. Hal ini dilakukan untuk mempermudah pengaturan. Persamaan-persamaan berikut akan disajikan dalam kerangka “*hybrid*”, atau H-frame. Persamaan (2.43) menunjukkan vektor kecepatan *quadcopter* yang digeneralisasi pada H-frame.

$$\dot{\xi} = [\Gamma^E \quad \omega^B]^T = [\dot{X} \quad \dot{Y} \quad \dot{Z} \quad p \quad q \quad r]^T \tag{2.43}$$

Dinamika sistem pada H-frame dapat dituliskan dalam bentuk matriks menurut Persamaan (2.44).

$$M_H \dot{\xi} + C_H(\xi) \xi = G_H + O_H(\xi) \Omega + E_H(\xi) \Omega^2 \tag{2.44}$$

Berikut ini akan ditentukan semua matriks dan vektor yang ditunjukkan oleh persamaan di atas. Matriks inersia sistem terhadap H-

*frame*  $\mathbf{M}_H$  sama dengan matriks inersia sistem terhadap *B-frame*, dan ditentukan seperti pada Persamaan (2.45).

$$\mathbf{M}_H = \mathbf{M}_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \quad (2.45)$$

Namun, matriks sentripetal Coriolis terhadap *H-frame*  $\mathbf{C}_H(\xi)$  tidak sama dengan matriks sentripetal Coriolis terhadap *B-frame* dan ditentukan seperti pada Persamaan (2.46).

$$\mathbf{C}_H(\xi) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I \omega_B) \end{bmatrix} \quad (2.46)$$

Vektor gravitasi terhadap *H-frame*  $\mathbf{G}_H$  dituliskan pada Persamaan (2.47). Dapat dilihat bahwa gravitasi mempengaruhi ketiga persamaan linear, namun lebih berpengaruh terhadap ketinggian *quadcopter*.

$$\mathbf{G}_H = \begin{bmatrix} F_G^E \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.47)$$

Efek giroskopis yang dihasilkan oleh putaran *propeller* tidak berubah karena hanya mempengaruhi persamaan angular yang mengacu kepada *B-frame*. Oleh karena itu, matriks giroskopis terhadap *H-frame*  $\mathbf{O}_H(\xi)$  dibuat sama dengan Persamaan (2.48).

$$\mathbf{O}_H(\xi)\Omega = \mathbf{O}_B(v)\Omega = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \end{bmatrix} \Omega$$

$$\mathbf{O}_H(\xi)\Omega = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega \quad (2.48)$$

Matriks perpindahan terhadap H-frame  $\mathbf{E}_H(\xi)$  berbeda dengan perpindahan terhadap B-frame karena input  $U_1$  mempengaruhi semua persamaan linear dengan matriks rotasi  $\mathbf{R}_\theta$ . Perkalian produk antara matriks perpindahan dengan kuadrat kecepatan *propeller* ditunjukkan pada Persamaan (2.49).

$$\begin{aligned} \mathbf{E}_H(\xi)\Omega^2 &= \begin{bmatrix} \mathbf{R}_\theta & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{E}_B\Omega^2 \\ \mathbf{E}_H(\xi)\Omega^2 &= \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi)U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)U_1 \\ (c_\theta c_\phi)U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \end{aligned} \quad (2.49)$$

Dengan menyusun kembali Persamaan (2.44), dapat ditemukan rumus untuk turunan vektor kecepatan yang digeneralisasi terhadap H-frame yang dapat dilihat pada Persamaan (2.50).

$$\dot{\xi} = \mathbf{M}_H^{-1} (\mathbf{C}_H(\xi)\xi + \mathbf{G}_H + \mathbf{O}_H(\xi)\Omega + \mathbf{E}_H(\xi)\Omega^2) \quad (2.50)$$

#### 2.1.4 Model Matematika [2], [3]

Dari analisis kinematika dan dinamika diperoleh Persamaan model matematika dari *quadcopter* seperti pada Persamaan (2.51). Dengan melihat secara sederhana pada Persamaan (2.51), posisi pada sumbu Z, dan posisi sudut *roll*, *pitch*, *yaw* dapat dikontrol secara langsung, berturut-turut dengan menggunakan  $U_1$ ,  $U_2$ ,  $U_3$ , dan  $U_4$ . Kontrol pada posisi maju (X), dan menyamping (Y) dapat dilakukan dengan mengatur sudut *pitch* dan *roll* dengan syarat gaya angkat ( $U_1$ ) tidak sama dengan nol.

$$\begin{aligned}
\ddot{X} &= \frac{U_1}{m} (\cos \Psi \sin \theta \cos \phi + \sin \Psi \sin \phi) \\
\ddot{Y} &= \frac{U_1}{m} (\sin \Psi \sin \theta \cos \phi - \cos \Psi \sin \phi) \\
\ddot{Z} &= -g + \frac{U_1}{m} (\cos \theta \cos \phi) \\
\dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{I_r}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\
\dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_r}{I_{yy}} q\Omega + \frac{U_2}{I_{yy}} \\
\dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}}
\end{aligned} \tag{2.51}$$

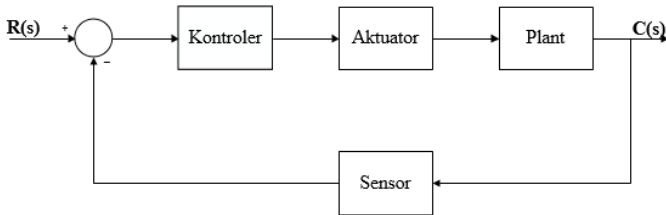
Nilai *input* dari *quadcopter* merupakan gaya angkat tiap *propeller*, yang telah dimodelkan secara teoritis dalam Persamaan (2.52), adalah sebagai berikut:

$$\begin{aligned}
U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
U_2 &= bl(-\Omega_2^2 + \Omega_4^2) \\
U_3 &= bl(-\Omega_1^2 + \Omega_3^2) \\
U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\
\Omega &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4
\end{aligned} \tag{2.52}$$

## 2.2 Kontroler

Dalam sebuah sistem, kontroler memiliki kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Dengan artian bahwa karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu *sub*-sistem, yaitu kontroler. Salah satu tugas komponen kontroler adalah meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal referensi dan sinyal aktual. Hal ini sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang sama dengan sinyal referensi.

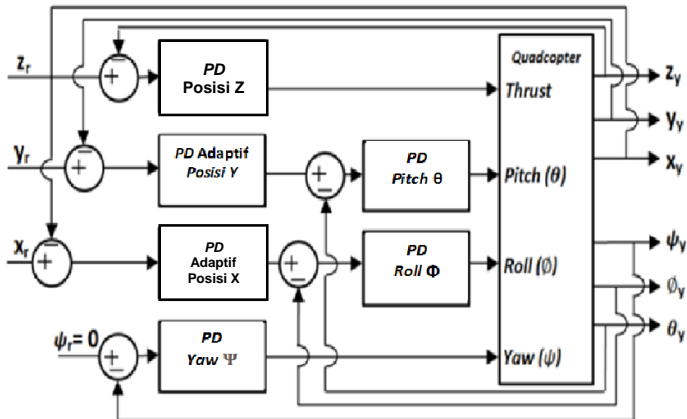
Semakin cepat reaksi sistem (sinyal aktual) mengikuti sinyal referensi dan semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan. Letak kontroler dapat dilihat pada blok diagram *closed loop* pada Gambar 2.10. Namun hal tersebut dapat bervariasi sesuai dengan kebutuhan desain.



**Gambar 2.10** Blok Diagram *Closed Loop*

Pada penelitian ini digunakan kontroler *Proportional Derivative* (PD) yang digunakan untuk pengendalian gerak rotasi atau *inner loop*. Sedangkan untuk pengendalian gerak translasi atau *outer loop* digunakan kontroler Adaptif.

Adapun blok diagram kontroler *quadcopter* yang dirancang adalah sebagai berikut:



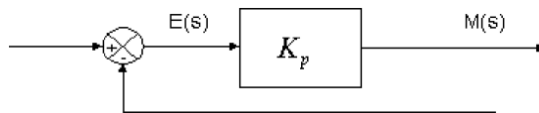
**Gambar 2.11** Skema Blok Diagram Rancangan Sistem

### 2.2.1 Proportional Derivative (PD) [4]

Sebuah kontroler proporsional ditambah integral ditambah derivatif (PD) adalah salah satu mekanisme umpan balik yang banyak digunakan dalam sistem pengaturan industri. Sebuah kontroler PD menghitung nilai kesalahan sebagai perbedaan antara variabel proses terukur dan set point yang diinginkan. Hingga saat ini, PD dapat dikatakan sebagai kontroler terbaik karena tidak diperlukan teknik yang rumit dalam tahap perancangan. Namun, parameter PD yang digunakan dalam penghitungan harus diatur sesuai dengan sifat sistem. Berikut adalah penjelasan dari kontroler *Proportional* (P) dan *Derivative* (D):

#### 2.2.1.1 Proportional (P)

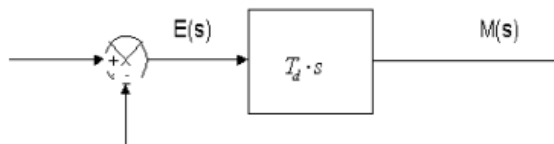
*P Controller* (Kontroler Proporsional) memiliki keluaran yang sebanding atau proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan bahwa keluaran pengontrol proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengeluarkan output sinyal sebesar konstanta pengalinya.



Gambar 2.12 Blok Diagram Kontroler Proporsional

#### 2.2.1.2 Derivative (P)

Keluaran pengontrol Derivative memiliki sifat seperti halnya suatu operasi differensial. Perubahan yang mendadak pada masukan pengontrol, akan mengakibatkan perubahan yang sangat besar dan cepat.



Gambar 2.23 Blok Diagram Kontroler Derivatif

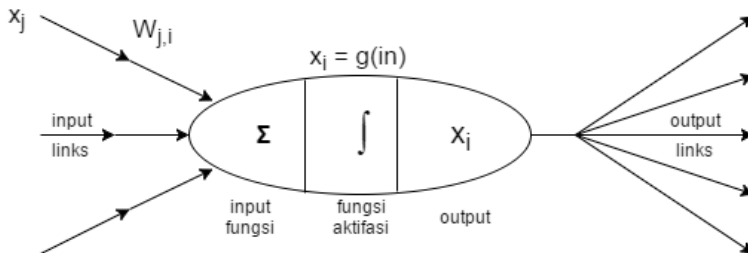
### 2.2.2 Kontroler Adaptif [5], [6], [7]

Kontroler adaptif merupakan suatu kontroler dengan algoritma yang mampu untuk merevisi nilai parameter atau struktur kontroler dengan cara mengikuti perubahan parameter dari struktur *plant* sehingga sistem yang dikendalikannya dapat beradaptasi untuk memenuhi spesifikasi desain yang ditentukan. Kontroler Adaptif dibutuhkan apabila parameter *plant*-nya berubah-ubah sehingga menyebabkan perubahan spesifikasi respon yang cukup signifikan.

Kontroler Adaptif tidak dapat digunakan apabila terjadi kondisi-kondisi *plant* sebagai berikut:

- a. Apabila kecepatan perubahan parameter lebih tinggi dari kecepatan beradaptasi.
- b. Apabila terdapat komponen non-linear diskontinyu.
- c. Apabila parameter berubah sebagai fungsi waktu.

Adapun dalam Tugas Akhir ini, kontroler Adaptif yang digunakan adalah proses adaptasi yang menirukan proses *learning* dari Jaringan Saraf Tiruan (*Neural Network*).



**Gambar 2.14** Model Tiruan Sebuah *Neuron*

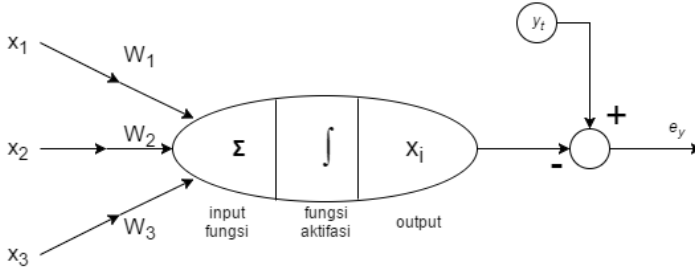
Dimana:  $x_j$  : Nilai aktivasi dari unit  $j$   
 $w_{j,i}$  : Bobot dari unit  $j$  ke  $i$   
 $in_i$  : Penjumlahan bobot dan masukan unit ke  $i$   
 $g$  : Fungsi aktivasi  
 $x_i$  : Nilai aktivasi dari unit  $i$

Misalkan ada sejumlah  $n$  sinyal masukan dan sejumlah  $n$  aktivasi, maka fungsi keluaran dari Neuron adalah sebagai berikut:



$$in_i = \sum_j W_{j,i} x_j \quad (2.53)$$

Dari Persamaan (2.53) dan Gambar (2.14), kita bisa menurunkan proses *learning* dari PD Adaptif dengan skema *back propagation* sebagai berikut:

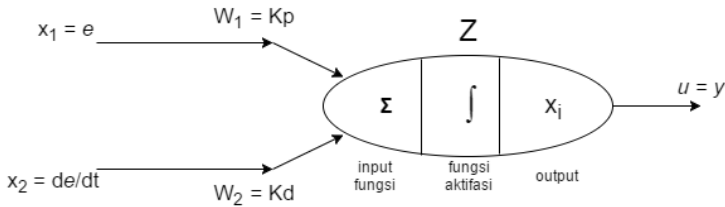


**Gambar 2.15** Skema *Back Propagation Learning*

Revisi bobot pada proses *back propagation* diperoleh sebagai berikut:

$$W_i^{Lama} = W_i^{Baru} + \alpha * e_y * f'(z) * x_i \quad (2.54)$$

Dimana  $\alpha$  merupakan *learning rate* dari proses *learning*. Dari Persamaan (2.54), bentuk PD Adaptif dapat dinyatakan dalam bentuk JST sebagai berikut :



**Gambar 2.16** Skema *Back Propagation Learning* dari PD Adaptif

Dimana didapat persamaan baru sebagai berikut:

$$Z = W_1 * X_1 + W_2 * X_2 = Kp * e + Kd \frac{de}{dt} \quad (2.55)$$

Dimisalkan  $y = \lambda Z$  ; Jika dipilih  $\lambda = 1$ , didapat

$$y = u = Z = Kp * e + Kd \frac{de}{dt} \quad (2.56)$$

$e_y$  atau  $e_u$  diperkirakan mendapatkan simpangan *error* terhadap *error model* dan dinyatakan oleh  $de$ . Dari Persamaan (2.54)-(2.56) didapat persamaan *back propagation learning* sebagai berikut:

$$Kp^{Baru} = Kp^{Lama} + \alpha * de * (1) * e \quad (2.57)$$

$$Kd^{Baru} = Kd^{Lama} + \alpha * de * (1) * \frac{de}{dt} \quad (2.58)$$

### 2.3 Linearisasi [8]

Suatu sistem yang kompleks memiliki hubungan antara besaran-besaran fisik dalam sistem bersifat nonlinear. Untuk menyelesaikan persoalan nonlinear, dapat digunakan dengan menggunakan pendekatan persamaan linear. Salah satu karakteristik dari sistem nonlinear adalah ketergantungan perilaku respon sistem pada besaran dan *input* dari sistem.

*Quadcopter* merupakan sistem yang memiliki hubungan antar variable bersifat nonlinear. Pada Persamaan (2.50) terlihat bahwa persamaan tersebut terdapat hubungan antar variable bersifat nonlinear karena terdapat fungsi trigonometri. Untuk dapat melakukan proses kontrol yang menggunakan kontroler linear yaitu PD dan PD adaptif, maka model matematika tersebut harus dilinearisasi. Teori linearisasi digunakan untuk mengatasi permasalahan sistem nonlinear agar dapat didekati dengan sistem linear.

Pada penelitian ini digunakan teknik linearisasi *feedback linearization* di mana merupakan salah satu teknik linearisasi dengan melakukan modifikasi pada variabel yang dikontrol sehingga dapat didekati dengan sistem linear. Misalkan suatu sistem mempunyai persamaan *state* seperti Persamaan (2.59). [4]

$$\begin{aligned} \dot{x} &= f(x) + G(x)u \\ y &= h(x) \end{aligned} \quad (2.59)$$

Domain tersebut mengandung *origin*, sehingga akan didapatkan sinyal kontrol modifikasi pada Persamaan (2.60).

$$u = \alpha(x) + \beta(x)v \quad (2.60)$$

*--halaman ini sengaja dikosongkan--*

## BAB III PERANCANGAN SISTEM

Pada Bab ini, perancangan sistem yang dilakukan dibagi menjadi empat tahapan. Pertama, parameter *quadcopter*. Kedua, proses linierisasi bentuk persamaan *state quadcopter* dengan metode *feedback linierization*, sehingga dapat dilakukan perancangan kontroler. Ketiga, perancangan kontroler *quadcopter* yang akan digunakan untuk melakukan *trajectory tracking quadcopter*. Terakhir, penyusunan simulasi sistem yang nantinya akan dibuat pada program Matlab R2015a.

### 3.1 Parameter *Quadcopter* [9]

Pada tugas akhir ini, digunakan parameter *quadcopter* miliki *quadcopter* Qball-X4. Parameter ini yang dijadikan sebagai nilai acuan perancangan kontroler dan simulasi. Nilai parameter dinyatakan dalam Tabel 3.1.

**Tabel 3.1** Parameter *Quadcopter*

No	Parameter	Simbol	Nilai
1	Massa	$m$	3,499 Kg
2	Gravitasi	$g$	9,81 Kg/m <sup>2</sup>
3	Momen inersia pada sumbu-X	$I_{xx}$	0,03 Kg.m <sup>2</sup>
4	Momen inersia pada sumbu-Y	$I_{yy}$	0,03 Kg.m <sup>2</sup>
5	Momen inersia pada sumbu-Z	$I_{zz}$	0,04 Kg.m <sup>2</sup>
6	Jarak rotor dari pusat massa	$l$	0,02 m
7	Gaya <i>drag</i>	$d$	$3,13 \times 10^{-5}$
8	Gaya <i>thrust</i>	$b$	$7,5 \times 10^{-7}$
9	<i>Bandwidth</i> actuator	$\omega$	15 rad/s
10	Konstanta gaya dorong	K	120 N

### 3.2 Linierisasi Persamaan *State Quadcopter*

Suatu sistem yang kompleks memiliki hubungan antara besaran-besaran fisik dalam sistem bersifat nonlinear. Untuk menyelesaikan persoalan nonlinear, dapat digunakan dengan menggunakan pendekatan persamaan linear. Salah satu karakteristik dari sistem nonlinear adalah ketergantungan perilaku respon sistem pada besaran dan *input* dari sistem.

*Quadcopter* merupakan sistem yang memiliki hubungan antar variabel bersifat nonlinear. Pada Persamaan (2.27) terlihat bahwa persamaan tersebut terdapat hubungan antar variable bersifat nonlinear karena terdapat fungsi trigonometri. Untuk dapat melakukan proses kontrol yang menggunakan kontroler linear yaitu PD dan adaptif, maka model matematika tersebut harus dilinearisasi. Teori linearisasi digunakan untuk mengatasi permasalahan sistem nonlinear agar dapat didekati dengan sistem linear.

Linierisasi persamaan *state* ini terdiri atas dua bagian, yaitu pada persamaan rotasi *quadcopter* dan persamaan translasi *quadcopter*.

### 3.2.1 Linierisasi Persamaan Gerak Rotasi

Proses linierisasi ini dilakukan pada persamaan *state* gerak rotasi *quadcopter*. Dari Persamaan (2.50) didapat persamaan *state* gerak rotasi sebagai berikut:

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{I_r}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \quad (3.1)$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_r}{I_{yy}} q\Omega + \frac{U_3}{I_{yy}} \quad (3.2)$$

$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \quad (3.3)$$

Dari Persamaan (3.1),(3.2),(3.3) dibuat permisalan konstanta sebagai berikut:

$$\dot{p} = U_2s ; \dot{q} = U_3s ; \dot{r} = U_4s \quad (3.4)$$

Adapun dibuat permisalan terhadap konstanta sebagai berikut:

$$\frac{I_{yy} - I_{zz}}{I_{xx}} = a_1 ; \frac{I_r}{I_{xx}} = a_2 ; \frac{1}{I_{xx}} = a_3 \quad (3.5)$$

$$\frac{I_{zz} - I_{xx}}{I_{yy}} = b_1 ; \frac{I_r}{I_{yy}} = b_2 ; \frac{1}{I_{yy}} = b_3 \quad (3.6)$$

$$\frac{I_{xx} - I_{yy}}{I_{zz}} = c_1 ; \frac{1}{I_{zz}} = c_2 \quad (3.7)$$

Persamaan (3.4),(3.5),(3.6),(3.7) dimasukkan kedalam Persamaan (3.1),(3.2),(3.3) sehingga menjadi persamaan berikut:

$$U2s = (a_1)qr - (a_2)p\Omega + (a_3)U_2 \quad (3.8)$$

$$U3s = (b_1)pr + (b_2)q\Omega + (b_3)U_3 \quad (3.9)$$

$$U4s = (c_1)pq + (c_2)U_4 \quad (3.10)$$

Persamaan (3.8),(3.9),(3.10) dilinierisasikan menjadi bentuk persamaan berikut:

$$U_2 = \frac{1}{a_3} (U2s + (a_2)p\Omega - (a_1)qr) \quad (3.11)$$

$$U_3 = \frac{1}{b_3} (U3s + (b_2)q\Omega - (b_1)pr) \quad (3.12)$$

$$U_4 = \frac{1}{c_2} (U4s - (c_1)pq) \quad (3.13)$$

### 3.2.2 Linierisasi Persamaan Gerak Translasi

Proses linierisasi ini dilakukan pada persamaan *state* gerak translasi *quadcopter*. Dari Persamaan (2.50) didapat persamaan *state* gerak translasi sebagai berikut:

$$\ddot{X} = \frac{U_1}{m} (\cos \Psi \sin \theta \cos \phi + \sin \Psi \sin \phi) \quad (3.14)$$

$$\ddot{Y} = \frac{U_1}{m} (\sin \Psi \sin \theta \cos \phi - \cos \Psi \sin \phi) \quad (3.15)$$

$$\ddot{Z} = -g + \frac{U_1}{m} (\cos \theta \cos \phi) \quad (3.16)$$

Dari Persamaan (3.14),(3.15),(3.16) dibuat permisalan konstanta sebagai berikut:

$$Uxs = \ddot{X} ; Uys = \ddot{Y} ; Uys = \ddot{Z} \quad (3.17)$$

$$Uy = \sin \theta ; Ux = \sin \Phi \quad (3.18)$$

$$\frac{\sin \Psi \sin \theta}{\cos \theta} = \tan \beta ; \beta = \tan^{-1} \frac{\sin \Psi \sin \theta}{\cos \theta} \quad (3.19)$$

Persamaan (3.17),(3.18) dimasukkan kedalam Persamaan (3.14),(3.15),(3.16).

$$Uy = \frac{1}{\cos \Psi \cos \Phi} \left( \frac{m}{U_1} Uxs - \sin \Psi \sin \Phi \right) \quad (3.20)$$

$$Uys = \frac{U_1}{m} \cos \Psi \left( \frac{\sin \Psi \sin \theta}{\cos \theta} \cos \Phi - \sin \Phi \right) \quad (3.21)$$

Dari Persamaan (3.21) berlaku rumus trigonometri:

$$\sin(\Phi - \beta) = \sin \Phi \sin \beta - \cos \Phi \cos \beta \quad (3.22)$$

Persamaan (3.22) menjadi

$$\frac{-1}{\cos \Phi} \sin(\Phi - \beta) = (\tan \beta \cos \Phi - \sin \Phi) \quad (3.23)$$

Masukkan Persamaan (3.23) dan (3.20) ke dalam Persamaan (3.21) sehingga menjadi.

$$Uys = \frac{U_1}{m} \cos \Psi \left( \frac{-1}{\cos \beta} \sin(\Phi - \beta) \right) \quad (3.24)$$

Dari Persamaan (3.20) dan (3.24), didapat nilai  $\theta$  dan  $\Phi$  yang sudah linier sebagai berikut:

$$\theta = \sin^{-1} Uy ; \Phi = \sin^{-1} \left( \frac{m - \cos \beta}{U_1 \cos \Psi} \right) + \beta \quad (3.25)$$

Untuk Uzs, didapat hasil linierisasi sebagai berikut:

$$U_1 = \frac{m(Uzs+g)}{\cos \theta \cos \Phi} \quad (3.26)$$



### 3.3 Perancangan Kontroler PD

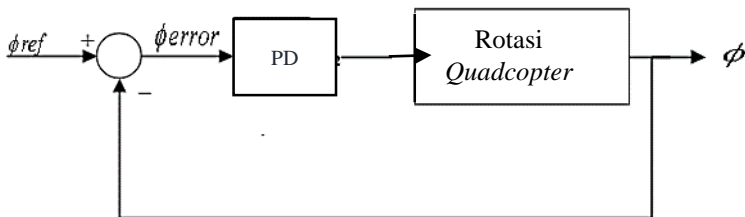
PD digunakan sebagai kontroler *quadcopter* baik pada bagian rotasi maupun translasi. Dimana gerak rotasi *quadcopter* terdiri dari gerak *roll*, *pitch*, dan *yaw*. Sedangkan Gerak translasi terdiri dari 3 macam, yaitu gerak translasi pada sumbu X, gerak translasi pada sumbu Y, dan gerak translasi pada sumbu Z Untuk mendesain kontroler PD diperlukan linearisasi, karena *quadcopter* adalah *plant* nonlinear. Metode linearisasi yang digunakan adalah *feedback linearization*. Untuk tuning parameter  $K_p$ ,  $K_d$  pada PD digunakan metode *MIT Procedures* yang memiliki langkah-langkah sebagai berikut:

1. *Tuning*  $K_p$  hingga sistem *steady-state* paling stabil dengan osilasi paling minimum.
2. *Tuning*  $K_i$  hingga nilai *error steady state* paling mendekati 0.
3. *Tuning*  $K_d$  untuk menghilangkan *overshoot* (hingga *overshoot* minimum).

Karena kontroler ini hanya menggunakan kontroler PD, maka hanya diambil langkah satu dan tiga dari *MIT Procedure* untuk menentukan nilai  $K_p$  dan  $K_d$ .

#### 3.8.1 Perancangan Kontroler PD Sudut Roll

Pada perancangan kontroler sudut *roll* digunakan Persamaan (3.11). Sudut *roll* berarti akan mempengaruhi pergerakan *quadcopter* pada sumbu Y. Blok diagram pengontrolan sudut *roll* ditunjukkan pada Gambar 3.3 dengan  $\phi_{ref}$  adalah sinyal kontrol dari kontroler translasi sumbu Y. Hal tersebut dikarenakan gerak translasi dikontrol secara *cascade*.



Gambar 3.1 Blok Diagram Pengendalian Sudut Roll

Untuk perancangan PD sudut *roll*, langkah pertama ialah mendefinisikan hubungan linear antar *variable* pada model matematika sudut *roll*. Hubungan linear tersebut terdapat pada Persamaan (3.27)-(3.29) dan nilai parameter *quadcopter* dari Tabel 3.1.

$$\dot{\phi} = p \quad (3.27)$$

$$\ddot{\phi} = \dot{p} \quad (3.28)$$

$$\ddot{\phi} = -0,3333 \, qr - 0,0016 \, q\Omega + 33,3333 \, U_2 \quad (3.29)$$

Menggunakan kontroler PD, berarti harus mencari nilai Kp, dan Kd. Pada pengaturan gerak rotasi *quadcopter* ini digunakan metode *try and error* dan didapatkan nilai Kp, dan Kd untuk pengaturan sudut *roll* pada Tabel 3.2.

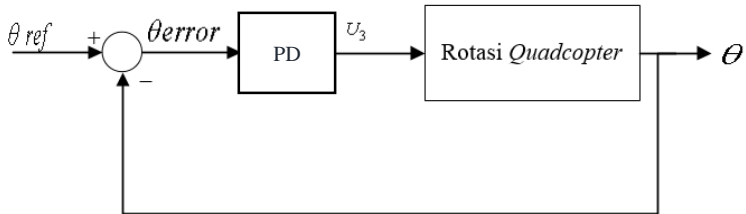
**Tabel 3.2** Nilai Kp dan Kd Kontroler PD Sudut *Roll*

No	Parameter	Nilai
1	Kp	1
2	Kd	30

Dari Tabel 3.2 dapat dilihat bahwa nilai Kp bernilai 1. Kp sengaja bernilai kecil karena ditakutkan bila hasil simulasi diimplementasikan ke *plant quadcopter*, *plant* tidak sanggup bekerja sesuai nilai Kp yang diinginkan. Dikarenakan nilai Kp yang kecil, respon tidak terlalu cepat, namun sesuai dengan kondisi *plant*. Selain itu, *overshot* yang dihasilkan juga tidak terlalu besar. Untuk mengantisipasi hal tersebut maka nilai Kd juga yang relative kecil, yaitu bernilai 30.

### 3.8.2 Perancangan Kontroler PD Sudut *Pitch*

Pada perancangan kontroler sudut *pitch* digunakan Persamaan (3.12). Sudut *roll* berarti akan mempengaruhi pergerakan *quadcopter* pada sumbu Y. Blok diagram pengontrolan sudut *roll* ditunjukkan pada Gambar 3.3 dengan  $\phi_{ref}$  adalah sinyal kontrol dari kontroler translasi sumbu Y. Hal tersebut dikarenakan gerak translasi dikontrol secara *cascade*.



**Gambar 3.2** Blok Diagram Pengendalian Sudut *Pitch*

Pengontrolan gerak rotasi sudut *pitch* menggunakan Persamaan (3.12). Sudut *pitch* akan mengakibatkan *quadcopter* bergerak pada sumbu X. Blok diagram pengendalian sudut *pitch* ditunjukkan oleh Gambar 3.2 di mana  $\theta_{ref}$  adalah sinyal kontrol dari kontroler translasi pada sumbu X.

Hubungan antar variabel pada model matematika gerak rotasi sudut *pitch* yang terdapat pada Persamaan (3.30)-(3.32).

$$\dot{\theta} = q \quad (3.30)$$

$$\ddot{\theta} = \dot{q} \quad (3.31)$$

$$\ddot{\theta} = -0,3333 \, pr - 0,0016 \, p\Omega + 33,3333 \, U_3 \quad (3.32)$$

**Tabel 3.3** Nilai Kp dan Kd Kontroler PD Sudut *Pitch*

No	Parameter	Nilai
1	Kp	1
2	Kd	30

Dari Tabel 3.3 dapat dilihat bahwa nilai Kp bernilai 1. Kp sengaja bernilai kecil karena ditakutkan bila hasil simulasi diimplementasikan ke *plant quadcopter*, *plant* tidak sanggup bekerja sesuai nilai Kp yang diinginkan. Dikarenakan nilai Kp yang kecil, respon tidak terlalu cepat, namun sesuai dengan kondisi *plant*. Selain itu, *overshot* yang dihasilkan juga tidak terlalu besar. Untuk mengantisipasi hal tersebut maka nilai Kd juga yang relative kecil, yaitu bernilai 30.

### 3.8.3 Perancangan Kontroler PD Sudut *Yaw*

Perancangan PD pada pengendalian sudut *yaw* menggunakan Persamaan (3.13). Kontroler sudut *yaw* dirancang sebagai regulator, di mana sudut *yaw* akan dipertahankan pada sekitar 0 radian. Hubungan antar

*variable* pada model matematika gerak rotasi sudut *yaw* terdapat pada Persamaan (3.33)-(3.35)

$$\dot{\psi} = r \quad (3.33)$$

$$\ddot{\psi} = \dot{r} \quad (3.34)$$

$$\ddot{\psi} = 25 U_4 \quad (3.35)$$

**Tabel 3.4** Nilai Kp dan Kd Kontroler PD Sudut *Yaw*

No	Parameter	Nilai
1	Kp	1
2	Kd	30

### 3.3.4 Perancangan Kontroler PD Gerak *Heve* pada Sumbu-Z

Pada perancangan gerak *heve* pada sumbu-Z digunakan hasil linierisasi pada Persamaan (3.26). Gerak *heve* pada sumbu-Z inilah yang mengatur ketinggian *quadcopter* saat melakukan *tracking* pada lintasan sumbu-X dan sumbu-Y. Blok diagram pengontrolan gerak *heve* pada sumbu-Z ditunjukkan pada Gambar (2.11).

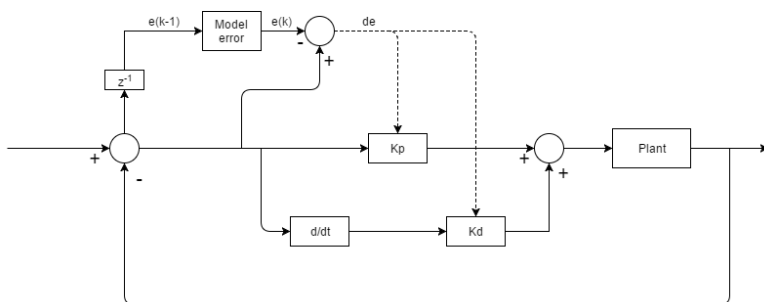
Adapun nilai Kp dan Kd yang digunakan dalam kontroler PD ini adalah sebagai berikut:

**Tabel 3.5** Nilai Kp dan Kd Kontroler PD Gerak *Heve* pada Sumbu-Z

No	Parameter	Nilai
1	Kp	1
2	Kd	20

## 3.4 Perancangan Kontroler PD Adaptif

PD (*Proportional Derivative*) adaptif digunakan untuk pengendalian gerak translasi. PD ini memiliki nilai Kp dan Kd yang selalu berubah dengan cara beradaptasi pada nilai *error* posisi pada sistem. Adapun kontroler PD adaptif ini memiliki skema sebagai berikut:

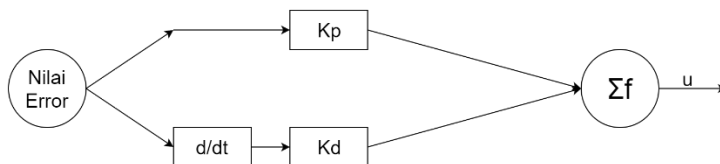


**Gambar 3.3** Skema Blok Diagram PD Adaptif

Kontroler PD sebagai suatu Jaringan Saraf Tiruan (JST) dengan persamaan sebagai berikut:

$$u = Kp * e + Kd * \frac{de}{dt} \quad (3.35)$$

Dalam bentuk JST, Persamaan (3.35) dapat digambarkan sebagai berikut:



**Gambar 3.4** Skema Jaringan Saraf Tiruan Kontroler PD Adaptif

Dari skema blok diagram diatas, didapat algoritma proses *learning* nilai Kp dan Kd dengan langkah-langkah sebagai berikut:

1. Inisialisasi nilai Kp, Kd,  $\alpha$  dan Ku. Dimana nilai Kp dan Kd inisialisasi didapat melalui MIT *procedures*. Adapun nilai  $\alpha$  merupakan *learning rate* dari kontroler PD adaptif. Ku merupakan konstanta pengali yang nantinya digunakan untuk mencari nilai *error* dari *output plant* atau u. Berikut adalah nilai-nilai inisialisasi dari Kp, Kd,  $\alpha$  dan Ku pada kontroler PD adaptif gerak translasi posisi X dan posisi Y.

**Tabel 3.6** Parameter Kontroler PD Adaptif Gerak Translasi Pada Sumbu X

No	Parameter	Nilai
1	Kp	10
2	Kd	20
3	$\alpha$	1
4	Ku	0,005

**Tabel 3.7** Parameter Kontroler PD Adaptif Gerak Translasi Pada Sumbu X

No	Parameter	Nilai
1	Kp	10
2	Kd	20
3	$\alpha$	1
4	Ku	0,005

## 2. Perhitungan

- Mengukur *error* sistem  $e(k)$  dan  $e(k-1)$ .
- Menghitung *model error*.

$$e_u = e^{\frac{-\tau}{\tau}} * (k - 1) \quad (3.36)$$

- Menghitung perbedaan *error* sistem dengan *model error*.

$$de = e_m(k) - e(k) \quad (3.37)$$

- Perhitungan *model error* (berbanding lurus dengan  $de$ ).

$$e_u = Ku * de \quad (3.38)$$

## 3. Proses revisi nilai Kp dan Kd

$$Kp_{baru} = Kp_{lama} + \alpha * e_u * e \quad (3.39)$$

$$Kd_{baru} = Kd_{lama} + \alpha * e_u * \frac{de}{dt} \quad (3.40)$$

## 4. Perhitungan sinyal control

$$u = Kp * e + Kd \frac{de}{dt} \quad (3.41)$$

## 5. Ulangi langkah ke-2

Perhitungan nilai  $K_p$  dan  $K_d$  untuk PD adaptif ini dilakukan dengan menggunakan program Matlab R2015a. Dengan skema PD Adaptif ini, diharapkan kontroler PD mampu beradaptasi dengan nilai *error* saat melakukan gerak translasi *trajectory tracking*.

*--halaman ini sengaja dikosongkan--*



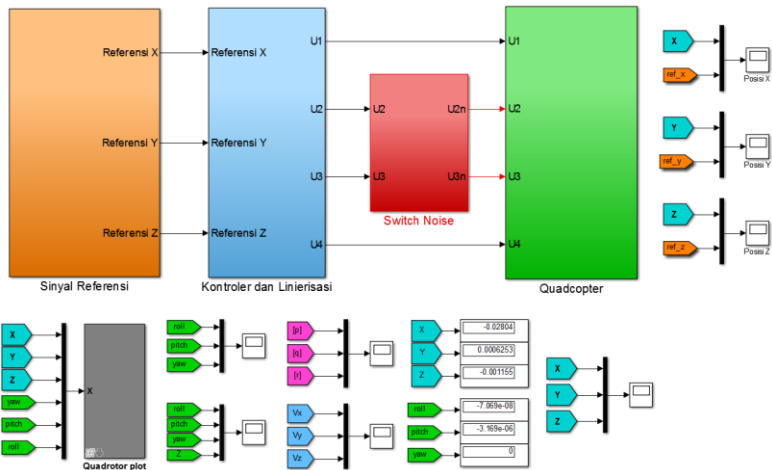
## BAB IV PENGUJIAN DAN ANALISIS

Berdasarkan spesifikasi dan perancangan sistem yang telah dijelaskan pada bab sebelumnya, maka dilakukan pengujian dan analisis terhadap perilaku sistem. Tujuan dari Pengujian adalah untuk mengetahui performansi sistem hasil perancangan. Jika performansi yang diharapkan belum sesuai, maka perlu diadakan analisis untuk penyempurnaan kinerja sistem sekaligus dapat digunakan untuk pengembangan lebih lanjut.

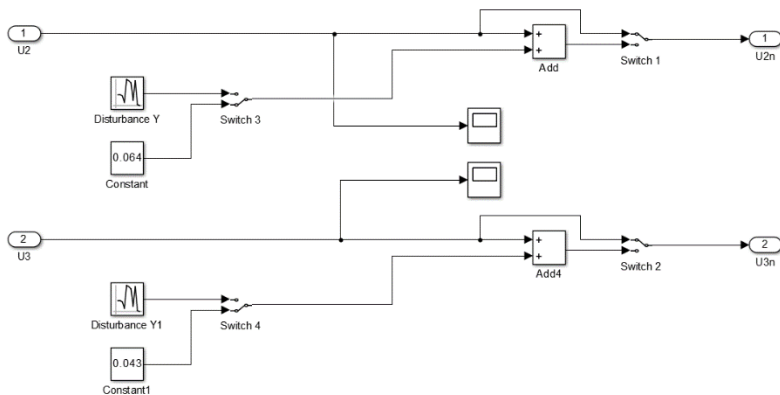
Pada bab ini akan dijelaskan hasil simulasi dan analisa dari perancangan gerak lateral *quadcopter* dengan menggunakan kontroler PD Adaptif dimana kontroler ini telah ditentukan nilai  $K_p$  dan  $K_d$  dengan menggunakan MIT *Procedures* pada bab sebelumnya.

### 4.1 Skema Simulasi Pergerakan *Quadcopter*

Perancangan kontroler untuk *trajectory tracking quadcopter* dirancang dengan PD dan PD Adaptif dengan menggunakan *software* Matlab R2015a. Pada simulasi ini *quadcopter* bergerak sesuai dengan skenario yang diinginkan yaitu *take-off* pada ketinggian tertentu, *way-point* dan *landing*. Gerak rotasi *quadcopter* tersebut dikontrol menggunakan kontroler PD untuk stabilisasi gerak *roll*, *pitch*, dan *yaw*, sedangkan untuk gerak translasi dikontrol dengan menggunakan kontroler PD Adaptif. Pengujian yang dilakukan meliputi pengujian tanpa *noise*, pengujian dengan menggunakan *contant noise* dengan tiga nilai *noise* yang berbeda, dan pengujian dengan menggunakan *random noise* dengan dua nilai berbeda. Hasil dari perancangan kontroler *quadcopter* ini dapat dilihat pada Gambar 4.1. Pengujian tanpa *noise* dan pengujian menggunakan *noise* dilakukan dengan memindah *switch* manual dalam sub-sistem *switch noise* pada simulasi yang terdapat pada Gambar 4.2.



**Gambar 4.1** Skema Simulink Simulasi Pergerakan *Quadcopter*



**Gambar 4.2** Blok Diagram Switch Manual

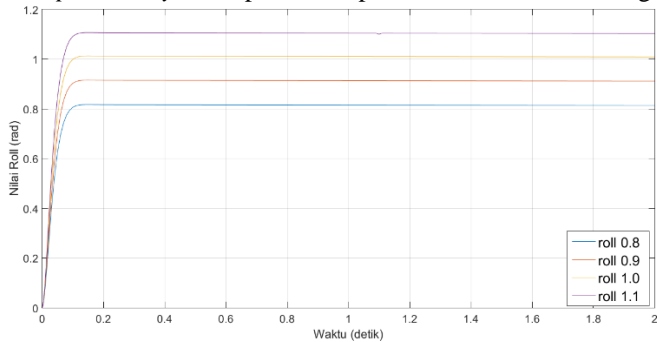
## 4.2 Simulasi Tanpa Noise

Pengujian tanpa *noise* dilakukan dengan mengganggu *quadcopter* berada pada kondisi ideal atau tanpa gangguan. Pengujian ini bermaksud untuk mengetahui performansi dari kontroler PD Adaptif yang telah dibuat. Pengujian dilakukan sebanyak tiga kali dengan parameter

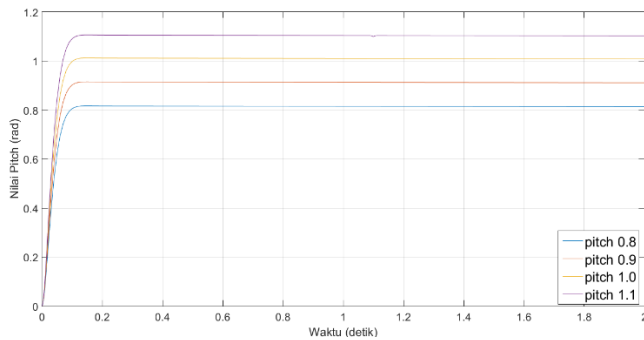
stabilisasi PD gerak *roll*, *pitch* dan *yaw* yang diubah nilainya. Perubahan parameter PD ini dilakukan untuk mendapatkan kestabilan pergerakan rotasi *roll*, *pitch*, dan *yaw* yang terbaik sehingga *quadcopter* dapat melakukan gerak translasi pada koordinat yang telah ditentukan.

#### 4.2.1 Simulasi Stabilisasi Gerak Rotasi *Quadcopter*

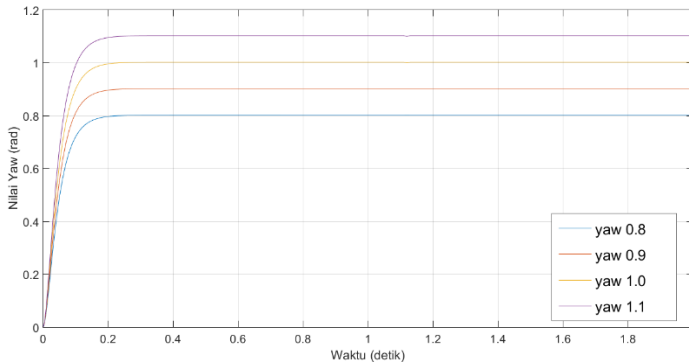
Saat melakukan gerak lateral pada sumbu-X dan sumbu-Y, pergerakan *quadcopter* sangat dipengaruhi oleh perubahan gerak rotasi sudut *roll*, *pitch* dan *yaw*. Pada Penelitian ini nilai stabilisasi *yaw* diberi masukan 0 sehingga orientasi *quadcopter* (*heading*) berada pada koordinat x(-). Oleh karena itu, stabilisasi gerak *roll*, *pitch* dan *yaw* sangatlah penting untuk mendapatkan pergerakan *quadcopter* seperti yang diharapkan. Hasil perancangan kontroler PD pada stabilisasi gerak *roll*, *pitch* dan *yaw* dapat dilihat pada Gambar 4.3-4.5 sebagai berikut:



Gambar 4.3 Stabilisasi Sudut *Roll*



Gambar 4.4 Stabilisasi Sudut *Pitch*



**Gambar 4.5** Stabilisasi Sudut Yaw

Berdasarkan grafik sudut respon pada Gambar 4.3-4.5, diketahui bahwa sudut *roll*, *pitch*, dan *yaw* berhasil distabilisasi dengan baik, karena respon sesuai dengan nilai referensi yang diberikan.

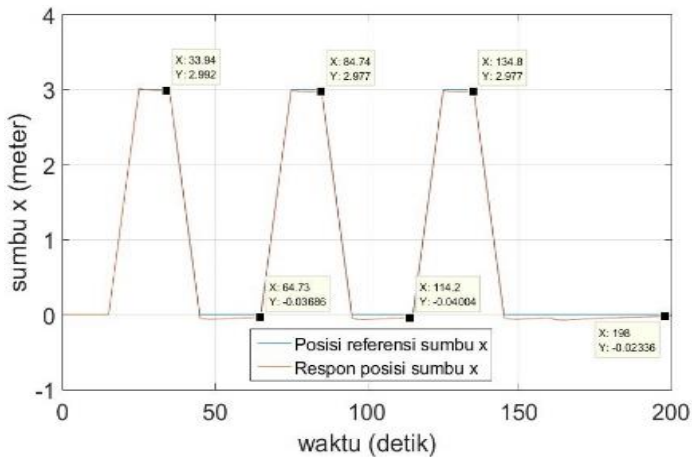
#### 4.2.2 *Trajectory Tracking Quadcopter*

Dalam *trajectory tracking quadcopter* ini, digunakan kontroler PD Adaptif untuk mengendalikan gerak translasi *quadcopter*. Adapun yang dikendalikan dengan menggunakan PD Adaptif ini hanya gerak lateral atau translasi pada sumbu-X dan sumbu-Y. Dalam desain PD Adaptif ini, nilai  $K_p$  dan  $K_d$  akan terus beradaptasi atau berubah-ubah menyesuaikan nilai *error* posisi pada gerak lateral atau translasi. Adapun prosedur dari pergerakan *quadcopter* adalah sebagai berikut:

1. Waktu total simulasi adalah 200 detik
2. *Quadcopter* melakukan *take-off* sejak detik nol hingga detik ke-5 menuju koordinat (0 0 3).
3. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 0 3) menuju (0 3 3).
4. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 3 3) menuju (3 3 3).
5. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 3 3) menuju (3 0 3).
6. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 0 3) menuju (0 0 3).
7. *Quadcopter* stabil pada koordinat (0 0 3) selama 10 detik.

8. *Quadcopter* mengulangi langkah kedua sampai ketujuh hingga dicapai detik ke-145.
9. *Quadcopter* stabil pada koordinat (0 0 3) selama 15 detik.
10. *Quadcopter* melakukan *landing* selama lima detik dari koordinat (0 0 3) menuju (0 0 0).
11. *Quadcopter* stabil pada koordinat (0 0 0) hingga detik ke-200.

Hasil pengujian respon pada *trajectory tracking* gerak translasi sumbu X, Y, dan Z dapat dilihat pada Gambar 4.6-4.9.



**Gambar 4.6** Respon Posisi Lintasan pada Sumbu-X tanpa *Noise*

Berdasarkan data pada Gambar 4.6, pada *tracking* pertama, diketahui bahwa saat langkah kedua hingga ketiga dilakukan, *quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama.

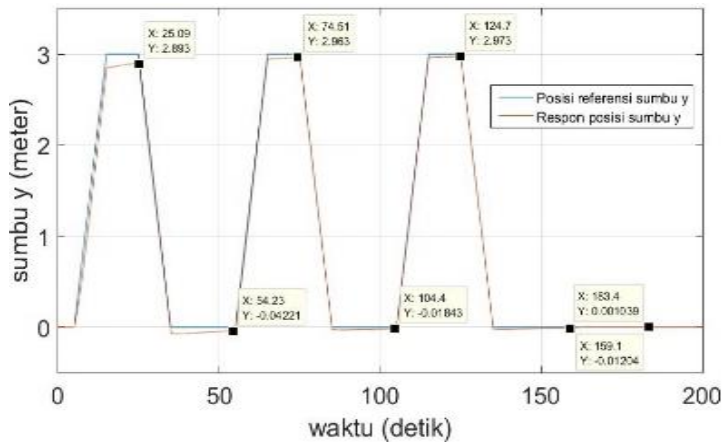
Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-X mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-X yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.1** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-X tanpa *Noise*

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	2,9921	2,9771	2,9776
2	-0,03666	-0,04004	-0,023336

Berdasarkan data pada Tabel 4.1, diketahui bahwa respon posisi memiliki nilai  $error \leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-X sudah sesuai dengan tujuan sistem yang diinginkan.



**Gambar 4.7** Respon Posisi Lintasan pada Sumbu-Y tanpa *Noise*

Berdasarkan data pada Gambar 4.7, untuk respon posisi *tracking* lintasan pada sumbu-Y, pada *tracking* pertama, diketahui bahwa saat

langkah kedua hingga ketiga dilakukan, *quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

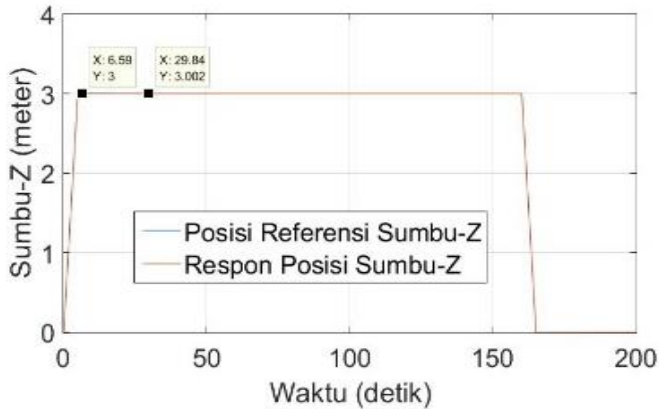
Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama. Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-Y mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-Y yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.2** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-Y tanpa *Noise*

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	2,893	2,963	2,973
2	-0,04221	-0,01843	-0,001039

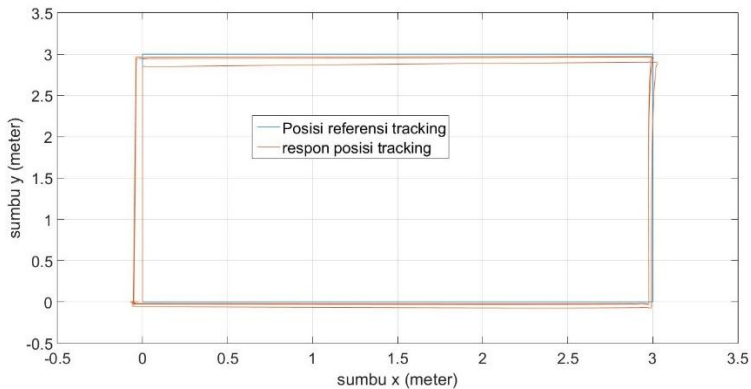
Berdasarkan data pada Tabel 4.2, diketahui bahwa respon posisi memiliki nilai *error*  $\leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-Y sudah sesuai dengan tujuan sistem yang diinginkan



**Gambar 4.8** Respon Posisi Lintasan pada Sumbu-Z tanpa *Noise*

Berdasarkan data pada Gambar 4.8, diketahui bahwa respon sinyal lintasan Z sangat mendekati sinyal referensi lintasan Z dengan error yang sangatlah kecil.

Adapun hasil dari *trajectory tracking quadcopter* ini dapat dilihat pada Gambar 4.9 dibawah



**Gambar 4.9** Posisi *Trajectory Tracking* tanpa *Noise*

Berdasarkan Gambar 4.9 diatas, dibuktikan bahwa *quadcopter* dapat mengikuti posisi referensi *tracking* dengan baik, walaupun terdapat

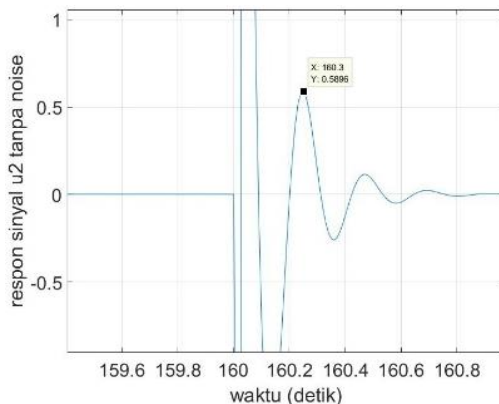


nilai  $error \leq 5\%$  antara posisi referensi dan respon posisinya. Dimana  $error$  posisi terbesar terjadi saat *quadcopter* melakukan *tracking* pada lintasan sumbu-X.

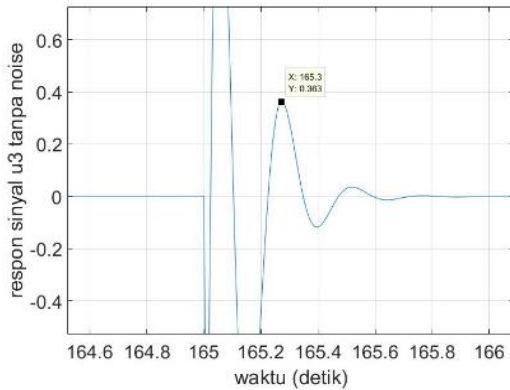
### 4.3 Simulasi Pengujian Dengan Noise

Simulasi pengujian dengan menggunakan *noise* dilakukan dengan mengikuti rute *tracking* yang sama seperti rute *tracking* pada simulasi tanpa *noise*, yaitu berupa gerak lateral atau translasi yang membentuk rute persegi pada koordinat sumbu-X dan sumbu-Y yang dijadikan posisi referensi. Adapun *noise* pada pengujian ini diberikan kepada gaya U2 dan U3, dimana *noise* ini merupakan suatu perumpamaan *disturbance* berupa angin yang mengganggu gerak *tracking quadcopter*.

Adapun batasan nilai *noise* yang diujikan pada gerak *tracking quadcopter* ini bernilai 1% dari *peak* terkecil pada sinyal U2 dan U3 yang didapat saat dilakukan simulasi *tracking* tanpa menggunakan *noise*. Nilai 1% didapat melalui metode *trial and error*, dimana pada saat metode ini dilakukan, *quadcopter* hanya mampu diberi nilai  $error \leq 1\%$ . Nilai *peak* terkecil dari sinyal U2 dan U3 dapat dilihat pada Gambar 4.10 dan 4.11, dimana nilai *peak* ini terjadi pada saat *quadcopter* melakukan gerakan *landing* ke koordinat asal (0 0 0) pada detik ke-160.



**Gambar 4.10** Sinyal U2 saat Dilakukan Simulasi tanpa Noise



**Gambar 4.11** Sinyal U3 saat Dilakukan Simulasi tanpa *Noise*

Dari Gambar 4.10 dan 4.11 diketahui bahwa *peak* sinyal U2 bernilai 0,59 dan *peak* sinyal U3 bernilai 0,363. Dari nilai kedua *peak* sinyal tersebut, didapat *noise* U2 dan U3 secara berurutan bernilai 0,0059 dan 0,00363.

#### 4.3.1 *Trajectory Tracking Quadcopter Dengan Noise Konstan*

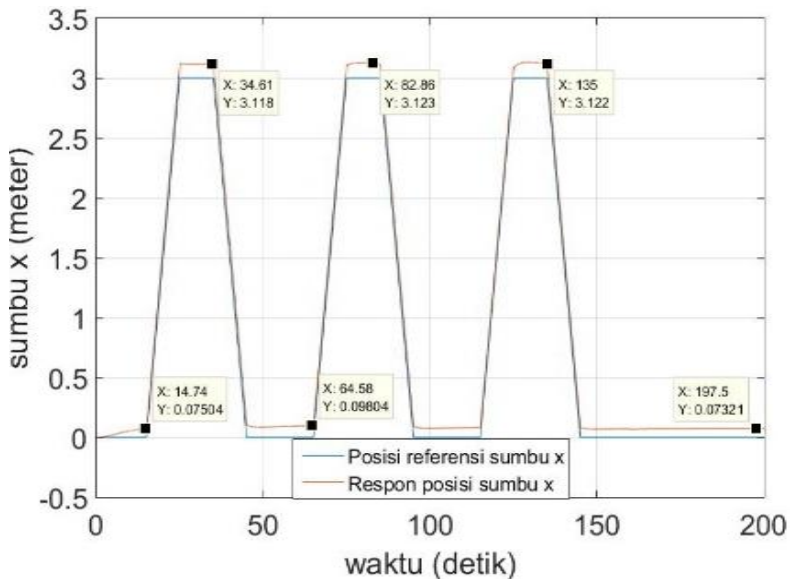
Sama seperti simulasi *trajectory tracking quadcopter* tanpa *noise*, digunakan kontroler PD Adaptif untuk mengendalikan gerak translasi *quadcopter*. Adapun yang dikendalikan dengan menggunakan PD Adaptif ini hanya gerak lateral atau translasi pada sumbu-X dan sumbu-Y. Dalam desain PD Adaptif ini, nilai  $K_p$  dan  $K_d$  akan terus beradaptasi atau berubah-ubah menyesuaikan nilai *error* posisi pada gerak lateral atau translasi.

Adapun prosedur dari pergerakan *quadcopter* dengan *noise* konstan ini mengikuti prosedur pergerakan *quadcopter* tanpa *noise* sebagai berikut:

1. Waktu total simulasi adalah 200 detik
2. *Quadcopter* melakukan *take-off* sejak detik nol hingga detik ke-5 menuju koordinat (0 0 3).
3. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 0 3) menuju (0 3 3).
4. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 3 3) menuju (3 3 3).

5. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 3 3) menuju (3 0 3).
6. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 0 3) menuju (0 0 3).
7. *Quadcopter* stabil pada koordinat (0 0 3) selama 10 detik.
8. *Quadcopter* mengulangi langkah kedua sampai ketujuh hingga dicapai detik ke-145.
9. *Quadcopter* stabil pada koordinat (0 0 3) selama 15 detik.
10. *Quadcopter* melakukan *landing* selama lima detik dari koordinat (0 0 3) menuju (0 0 0).
11. *Quadcopter* stabil pada koordinat (0 0 0) hingga detik ke-200.

Hasil simulasi dari pengujian *trajectory tracking quadcopter* dengan *noise* konstan dapat dilihat pada Gambar 4.12-4.15.



**Gambar 4.12** Respon Posisi Lintasan pada Sumbu-X dengan *Noise* Konstan

Berdasarkan data pada Gambar 4.12, pada *tracking* pertama, diketahui bahwa saat langkah kedua hingga ketiga dilakukan, *quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai

langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

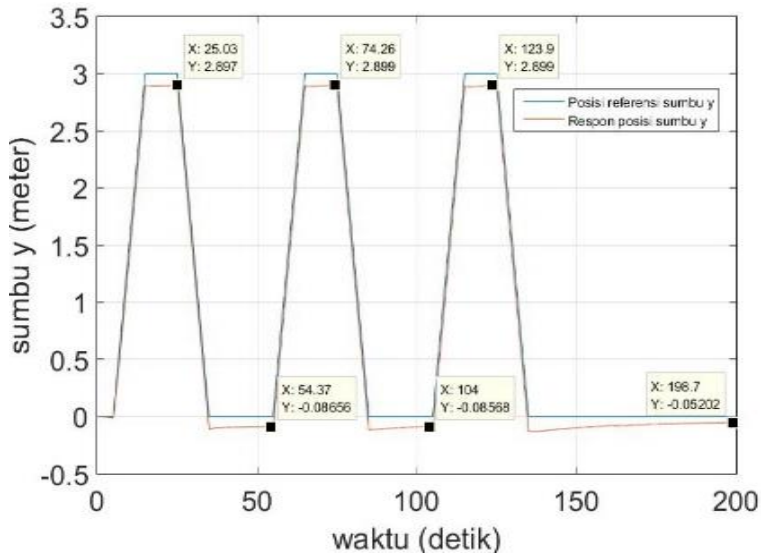
Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama. Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-X mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-X yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.3** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-X dengan *Noise* Konstan

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	3,118	3,123	3,122
2	0,09804	0,08216	0,07321

Berdasarkan data pada Tabel 4.3, diketahui bahwa respon posisi memiliki nilai *error*  $\leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-X sudah sesuai dengan tujuan sistem yang diinginkan.



**Gambar 4.13** Respon Posisi Lintasan pada Sumbu-Y dengan *Noise* Konstan

Berdasarkan data pada Gambar 4.13, untuk respon posisi *tracking* lintasan pada sumbu-Y, pada *tracking* pertama, diketahui bahwa saat langkah kedua hingga ketiga dilakukan, *quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

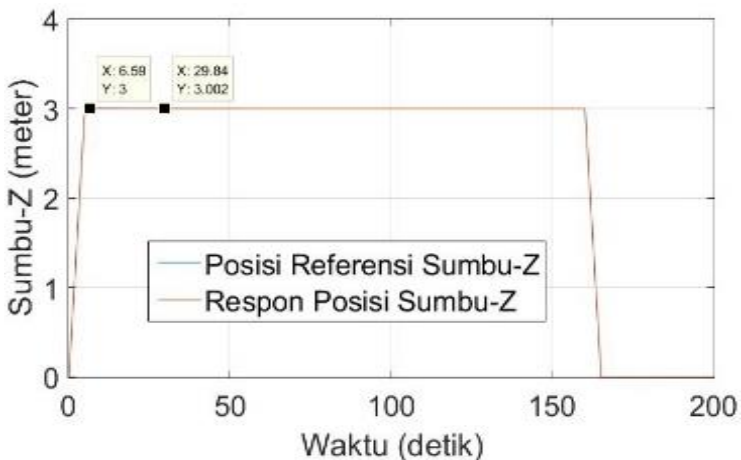
Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama. Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-Y mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-Y yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.4** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-Y dengan *Noise* Konstan

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	2,893	2,963	2,973
2	-0,04221	-0,01843	-0,001039

Berdasarkan data pada Tabel 4.4, diketahui bahwa respon posisi memiliki nilai  $error \leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-Y sudah sesuai dengan tujuan sistem yang diinginkan.

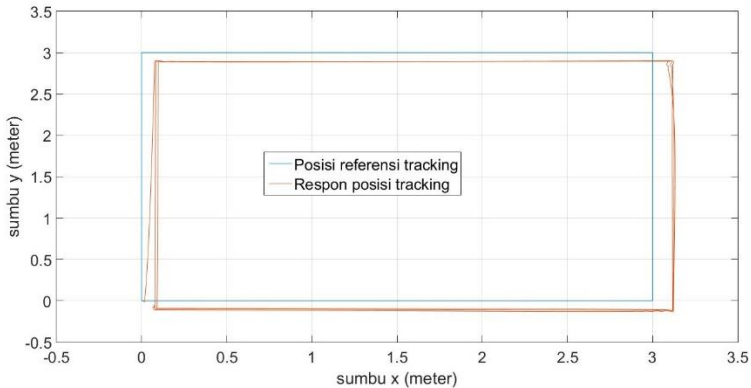


**Gambar 4.14** Respon Posisi Lintasan pada Sumbu-Z dengan *Noise* Konstan

Berdasarkan data pada Gambar 4.14, diketahui bahwa *quadcopter* mampu untuk tetap mampu menjaga ketinggiannya selama *tracking* dilakukan. Hal ini dibuktikan dengan respon posisi yang memiliki nilai *error* sangat teramat kecil dari referensi yang diberikan. Dimana *error* nya hanya melenceng 0,002 m dari posisi referensi yang diberikan.

Setelah melakukan analisa terhadap posisi di masing-masing lintasan pada sumbu-X, sumbu-Y, dan sumbu-Z. Selanjutnya dilakukan

analisa terhadap pergerakan *tracking quadcopter* dalam bentuk grafik-XY.



**Gambar 4.15** Posisi *Trajectory Tracking* dengan *Noise* Konstan

Berdasarkan data pada Gambar 4.15, diketahui bahwa *quadcopter* mampu mengikuti posisi referensi *tracking* dengan nilai *error* posisi yang kecil, dan *quadcopter* mampu beradaptasi dan meminimalisir nilai *error* posisi yang terjadi. Namun, proses adaptasinya tidak signifikan, sehingga nilai *error* posisi yang berkurang tiap kali *quadcopter* melakukan proses *tracking* sangatlah kecil.

#### 4.3.2 *Trajectory Tracking Quadcopter Dengan Random Noise*

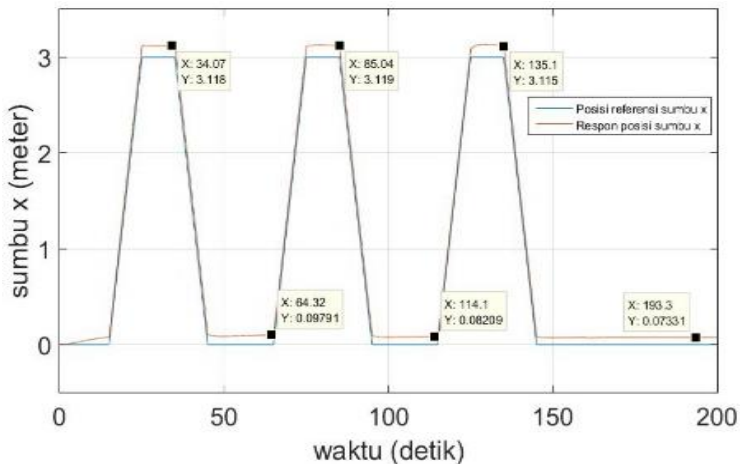
Sama seperti simulasi *trajectory tracking quadcopter* dengan *noise* konstan, digunakan kontroler PD Adaptif untuk mengendalikan gerak lateral atau translasi *quadcopter*. Adapun yang dikendalikan dengan menggunakan PD Adaptif ini hanya gerak lateral atau translasi pada sumbu-X dan sumbu-Y. Dalam desain PD Adaptif ini, nilai  $K_p$  dan  $K_d$  akan terus beradaptasi atau berubah-ubah menyesuaikan nilai *error* pada gerak lateral atau translasi.

Adapun prosedur dari pergerakan *quadcopter* dengan *random noise* ini mengikuti prosedur pergerakan *quadcopter* dengan *noise* konstan sebagai berikut:

1. Waktu total simulasi adalah 200 detik
2. *Quadcopter* melakukan *take-off* sejak detik nol hingga detik ke-5 menuju koordinat (0 0 3).

3. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 0 3) menuju (0 3 3).
4. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (0 3 3) menuju (3 3 3).
5. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 3 3) menuju (3 0 3).
6. *Quadcopter* melakukan perpindahan koordinat selama 10 detik dari (3 0 3) menuju (0 0 3).
7. *Quadcopter* stabil pada koordinat (0 0 3) selama 10 detik.
8. *Quadcopter* mengulangi langkah kedua sampai ketujuh hingga dicapai detik ke-145.
9. *Quadcopter* stabil pada koordinat (0 0 3) selama 15 detik.
10. *Quadcopter* melakukan *landing* selama lima detik dari koordinat (0 0 3) menuju (0 0 0).
11. *Quadcopter* stabil pada koordinat (0 0 0) hingga detik ke-200.

Hasil dari simulasi pengujian *trajectory tracking quadcopter* dengan *random noise* dapat dilihat pada Gambar 4.16-4.19.



**Gambar 4.16** Respon Posisi Lintasan pada Sumbu-X dengan *Random Noise*

Seperti kasus *trajectory tracking* lintasan dengan konstan *noise* pada sumbu-X. Berdasarkan data pada Gambar 4.16, pada *tracking* pertama, diketahui bahwa saat langkah kedua hingga ketiga dilakukan,



*quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

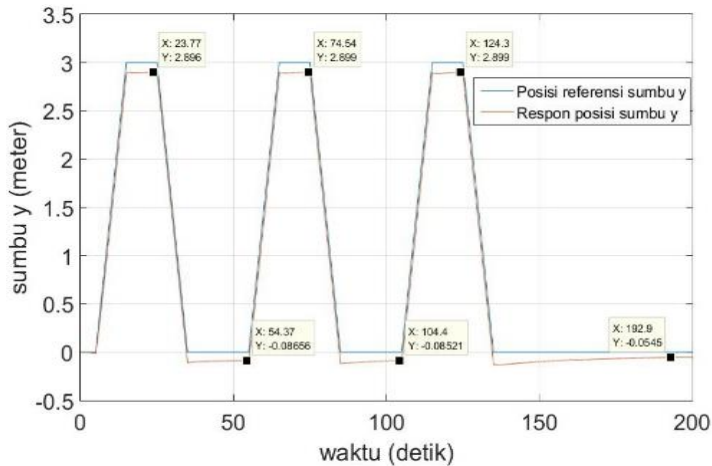
Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama. Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-X mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-X yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.5** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-X dengan *Random Noise*

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	3,118	3,119	3,115
2	0,07458	0,09791	0,07331

Berdasarkan data pada Tabel 4.5, diketahui bahwa respon posisi memiliki nilai *error*  $\leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-X sudah sesuai dengan tujuan sistem yang diinginkan.



**Gambar 4.17** Respon Posisi Lintasan pada Sumbu-Y dengan *Random Noise*

Berdasarkan data pada Gambar 4.17, untuk respon posisi *tracking* lintasan pada sumbu-Y, pada *tracking* pertama, diketahui bahwa saat langkah kedua hingga ketiga dilakukan, *quadcopter* mampu mengikuti posisi referensi dengan baik. Namun, saat mencapai langkah keempat hingga kelima, respon posisi dari *quadcopter* tidak mampu mengikuti posisi referensi yang ditentukan, hal ini dibuktikan dengan respon posisi dari *quadcopter*. Saat langkah keenam dilakukan, *quadcopter* mampu mengikuti posisi referensi yang telah ditentukan. Namun, saat langkah ketujuh dilakukan, *quadcopter* kembali tak mampu mengikuti referensi yang ditentukan.

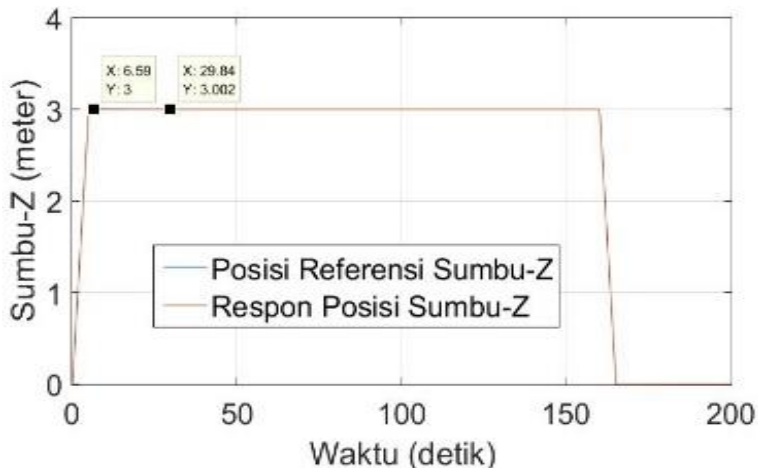
Respon yang sama ditunjukkan kembali saat dilakukan *tracking* kedua dan ketiga. Saat dilakukan *tracking* kedua, nilai *error* posisi menjadi lebih besar dari *error* posisi saat dilakukan *tracking* pertama. Namun, saat dilakukan *tracking* ketiga, nilai *error* yang terjadi pada saat *tracking* kedua dilakukan, mampu diminimalisir kontroler PD Adaptif *quadcopter* dengan nilai adaptasi yang kecil. Hal ini membuktikan bahwa kontroler PD Adaptif pada *tracking* gerak translasi pada sumbu-Y mampu bekerja dengan cukup baik, walaupun dengan respon adaptasi yang lambat.

Adapun data referensi dan respon posisi saat melakukan *tracking* gerak translasi pada sumbu-Y yang lebih presisi dapat dilihat pada tabel dibawah.

**Tabel 4.6** Respon Posisi *Tracking* Gerak Translasi pada Sumbu-Y dengan *Noise* Konstan

No	Respon Posisi <i>Tracking</i> Pertama	Respon Posisi <i>Tracking</i> Kedua	Respon Posisi <i>Tracking</i> Ketiga
1	2,896	2,899	2,899
2	-0,08656	-0,08521	-0,05754

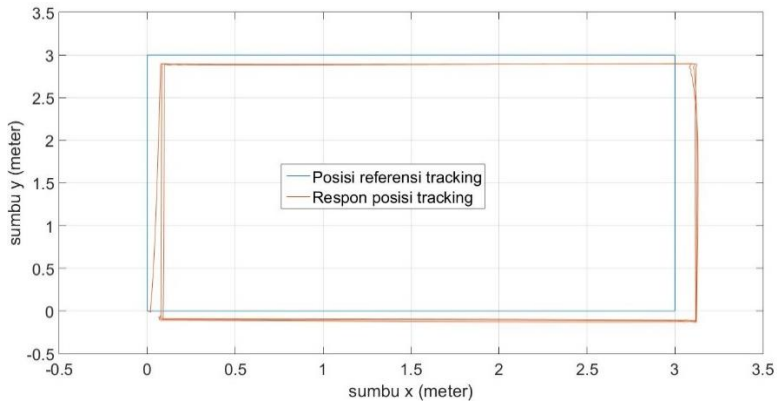
Berdasarkan data pada Tabel 4.6, diketahui bahwa respon posisi memiliki nilai  $error \leq 5\%$  posisi referensi yang diinginkan. Hal ini membuktikan bahwa kontroler PD Adaptif pada gerak translasi sumbu-Y sudah sesuai dengan tujuan sistem yang diinginkan.



**Gambar 4.18** Respon Posisi Lintasan pada Sumbu-Z dengan *Noise* Konstan

Berdasarkan data pada Gambar 4.18, diketahui bahwa *quadcopter* mampu untuk tetap mampu menjaga ketinggiannya selama *tracking* dilakukan. Hal ini dibuktikan dengan respon posisi yang memiliki nilai *error* sangat teramat kecil dari referensi yang diberikan. Dimana *error* nya hanya melenceng 0,002 m dari posisi referensi yang diberikan.

Setelah melakukan analisa terhadap posisi di masing-masing lintasan pada sumbu-X, sumbu-Y, dan sumbu-Z. Selanjutnya dilakukan analisa terhadap pergerakan *tracking quadcopter* dalam bentuk grafik-XY.



**Gambar 4.19** Posisi *Trajectory Tracking* dengan Random Noise

Sama seperti *trajectory tracking* dengan konstan *noise*, berdasarkan data pada Gambar 4.19, diketahui bahwa *quadcopter* mampu mengikuti posisi referensi *tracking* dengan nilai *error* posisi yang kecil, dan *quadcopter* mampu beradaptasi dan meminimalisir nilai *error* posisi yang terjadi. Namun, proses adaptasinya tidak signifikan, sehingga nilai *error* posisi yang berkurang tiap kali *quadcopter* melakukan proses *tracking* sangatlah kecil.

## **BAB V**

### **PENUTUP**

Hasil dari perancangan dan penelitian Tugas Akhir dirangkum dan dirumuskan kesimpulan. Kesimpulan ini menerangkan hasil dari pengujian dan simulasi yang telah dilaksanakan.

Selama proses perancangan dan penelitian, terdapat banyak kendala yang dihadapi. Kendala tersebut telah penulis rangkum dan dirumuskan dalam bentuk saran untuk penyempurnaan dan penelitian lebih lanjut

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut:

1. Kontroler PD untuk mengendalikan sudut *pitch*, *roll*, dan *yaw* pada gerak rotasi *quadcopter* bekerja dengan baik. Hal ini ditunjukkan dengan respon sudut yang sesuai dengan referensi sudut saat diberikan nilai-nilai pengujian berbeda pada masing-masing sudut.
2. Kontroler PD untuk mengendalikan gerak *Heve* pada lintasan sumbu-Z dapat bekerja dengan baik. Hal ini ditunjukkan dengan posisi *quadcopter* yang selalu stabil pada ketinggian tiga meter, dengan *error* bernilai 0,002 m atau sekitar 0,067% dari posisi referensi yang diinginkan.
3. Kontroler PD Adaptif untuk mengendalikan gerak lateral atau translasi pada sumbu-X dan sumbu-Y tanpa diberikan *noise* mampu bekerja dengan baik. Hal ini dibuktikan dengan nilai rata-rata *error* pada *tracking* pertama hingga ketiga bernilai  $\leq 5\%$ , hal ini dapat dilihat pada Tabel 4.1 dan 4.2.
4. Kontroler PD Adaptif untuk mengendalikan gerak lateral atau translasi pada sumbu-X dan sumbu-Y dengan diberikan *noise* konstan mampu bekerja dengan baik. Hal ini dibuktikan dengan nilai rata-rata *error* pada *tracking* pertama hingga ketiga bernilai  $\leq 5\%$ , hal ini dapat dilihat pada Tabel 4.3 dan 4.4.
5. Kontroler PD Adaptif untuk mengendalikan gerak lateral atau translasi pada sumbu-X dan sumbu-Y dengan diberikan *random noise* mampu bekerja dengan baik. Hal ini dibuktikan

dengan nilai rata-rata *error* pada *tracking* pertama hingga ketiga bernilai  $\leq 5\%$ , hal ini dapat dilihat pada Tabel 4.5 dan 4.6.

## 5.2 Saran

Dari hasil penelitian yang dilakukan, untuk pengembangan berikutnya, disarankan beberapa hal berikut ini:

1. Untuk menemukan nilai parameter  $K_p$  dan  $K_d$ , digunakan metode lain selain MIT *Procedures*, supaya dihasilkan nilai parameter  $K_p$  dan  $K_d$  yang lebih akurat.
2. Dapat digunakan metode adaptif lainnya untuk mendapatkan hasil simulasi yang lebih akurat.
3. Simulasi ini dapat dicoba untuk diimplementasikan pada *plant* riil *quadcopter* Quanser QBall-X4.

## DAFTAR PUSTAKA

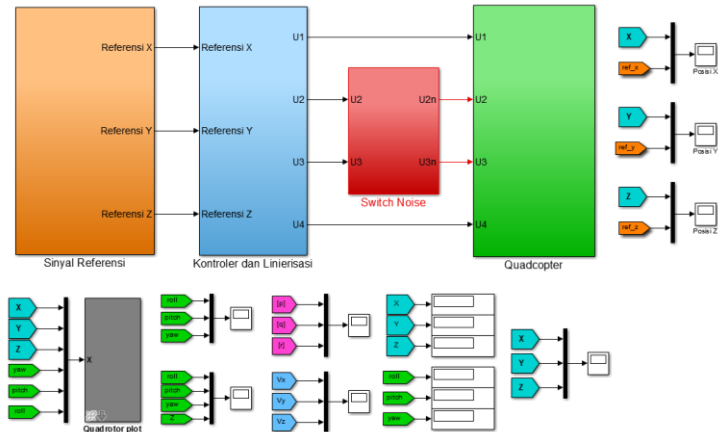
- [1] Tommaso Bresciani, “Modelling, Identification and Control of a *Quadcopter Helicopter*”. *Thesis*, Department of Automatic Control Lund University, 2008.
- [2] Spong, M.W, Vidyasagar, M. “*Robot Dynamic and Control*”, Prentice Hall, New Jersey, 2004.
- [3] Sulthoni, Mohammad Faizal., “Trajectory Tracking Gerak Cruise Pada Quadcopter Menggunakan Metode Linear Quadratic Tracking”. Surabaya: *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, 2016.
- [4] Ogata Katsuhiko, “*Modern Control Engineering 2<sup>nd</sup> Edition*”, Prentice Hall, New Jersey, 2008.
- [5] Åström, Karl Johan., Wittenmark Björk, “*Adaptive Control 2<sup>nd</sup> Edition*”, Pearson Education.
- [6] Iskandar Eka, “*Sistem Pengaturan Adaptif*”, Surabaya: Diktat, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember.
- [7] Rahmayanti, Farah Zakiyah., “*Jaringan Saraf Tiruan (JST)/ Neural Network (NN)*” Semarang: Diktat, Universitas Dian Nuswantoro.
- [8] H.K. Khalil and JW Grizzle, “*Nonlinear systems*”, Prentice Hall, New Jersey, 2002.
- [9] Sohendri Dinang., “Kontrol Tracking Fuzzy Berbasis Performa Robust Untuk Quadrotor”. Surabaya: *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, 2016.

*--halaman ini sengaja dikosongkan--*

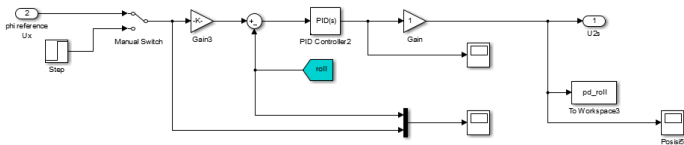


# LAMPIRAN A

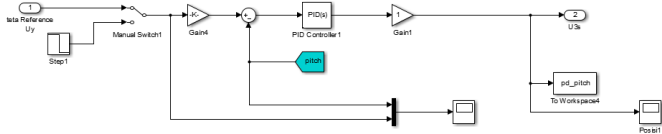
## A.1. Simulink Keseluruhan Sistem



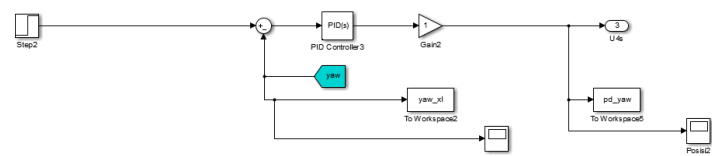
## A.2. Simulink Pengendalian Sudut Roll



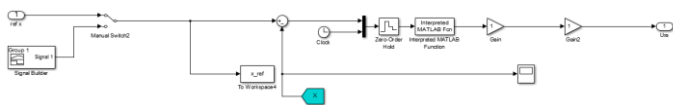
## A.3. Simulink Pengendalian Sudut Pitch



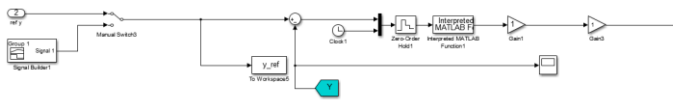
**A.4. Simulink Pengendalian Sudut Yaw**



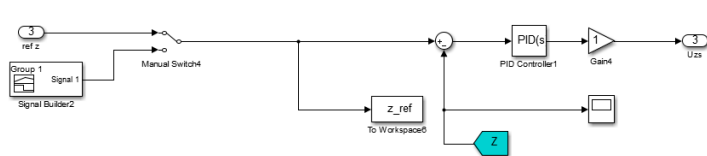
**A.5. Simulink Pengendalian Posisi X**



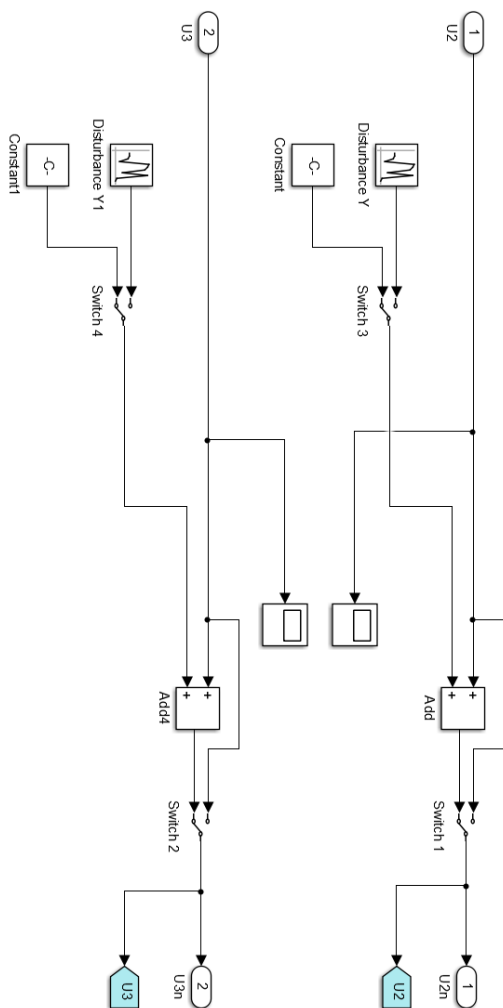
**A.6. Simulink Pengendalian Posisi Y**



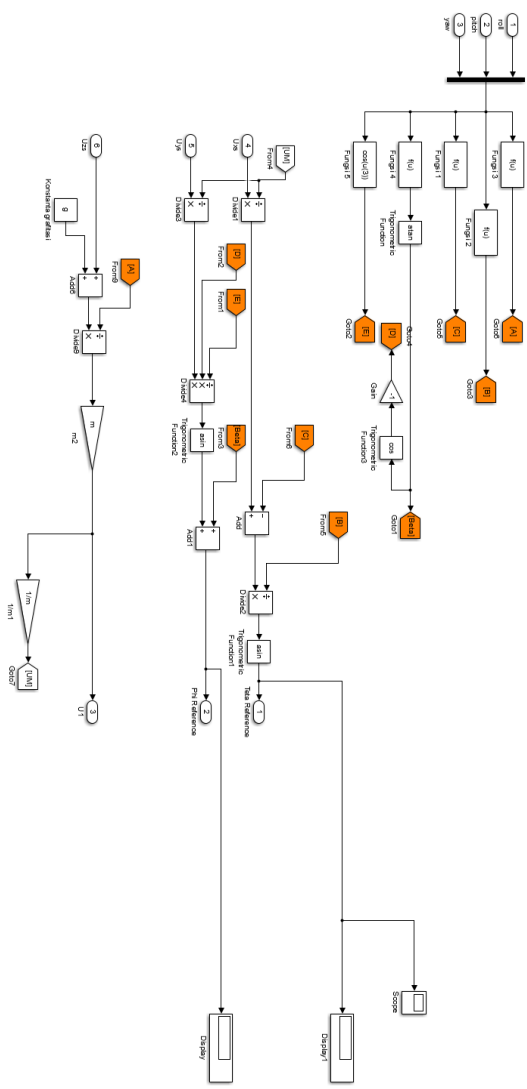
**A.7. Simulink Pengendalian Posisi Z**



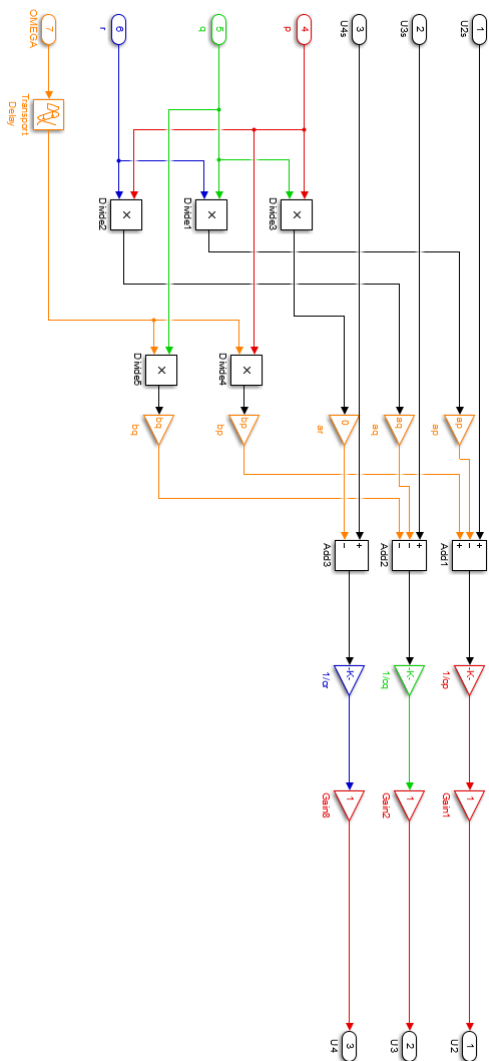
## A.8. Simulink Switch Manual untuk Noise



A.9. Simulink Linierisasi Persamaan Gerak Rotasi



### A.10. Simulink Linierisasi Persamaan Gerak Translasi



*--Halaman ini sengaja dikosongkan--*

## LAMPIRAN B

### B.1. Program PD Adaptif Posisi X pada Matlab

```
function ux=PDx(inx)
global kpx kdx exkm1 ts ku alp tau
ex=inx(1);
tt=inx(2);
if tt==0
    kpx=10;
    kdx=20;
    exkm1=0;
    ts=0.01;
    alp=0.005;
    ku=1;
    tau=2;
end
eu=ku*exp(-ts/tau)*ex;
for epp=1:20
    kpx=kpx+alp*eu*ex;
    kdx=kdx+alp*eu*ex;
end
dex=(ex-exkm1)/ts;
ux=kpx*ex + kdx*dex;
exkm1=ex;
kpdx = [kpx kdx]
end
```

### B.2. Program PD Adaptif Posisi X pada Matlab

```
function uy=PDy(iny)
global kpy kdy eykm1 ts ku alp tau
ey=iny(1);
tt=iny(2);
if tt==0
    kpy=10;
    kdy=20;
    eykm1=0;
    ts=0.01;
    alp=0.005;
    ku=1;
```

```

        tau=2;
    end
    eu=ku*exp(-ts/tau)*ey;
    for epp=1:20
        kpy=kpy+alp*eu*ey;
        kdy=kdy+alp*eu*ey;
    end
    dey=(ey-eykm1)/ts;
    uy=kpy*ey + kdy*dey;
    eykm1=ey;
    kpdy = [kpy kdy]
end

```



## **RIWAYAT PENULIS**



Benaverd Rizanda Putra Leihitu atau biasa dipanggil Ben lahir di Jakarta tanggal 11 Januari 1995. Ben merupakan anak pertama dari pasangan Erihanorizal dan Yolanda Leihitu. Lulus dari SDIT Nurul Hikmah pada tahun 2006, kemudian melanjutkan studi ke jenjang lebih lanjut di SMPN 1 Cibinong dan lulus pada tahun 2009. Kemudian melanjutkan ke SMAN 2 Cibinong dan lulus pada tahun 2012. Setelah menyelesaikan studi pada tingkat SMA, penulis melanjutkan ketingkat lebih lanjut, yaitu di Teknik Elektro ITS pada tahun 2012 dan memutuskan untuk mengambil bidang studi Teknik Sistem Pengaturan.

*--halaman ini sengaja dikosongkan--*